

Interaktive Metadateneditoren für den Einsatz bei geowissenschaftlichen Daten

Wolfgang Czegka¹, Stephan Braune², Knut Behrends³

1) Sächsische Akademie der Wissenschaften zu Leipzig, Karl-Tauchnitz-Str. 1,
D-04107 Leipzig, czegka@saw-leipzig.de

2) Alfred-Wegener-Institut für Polar- und Meeresforschung, Am Handelshafen
12, D-27570 Bremerhaven, sbraune@awi-bremerhaven.de

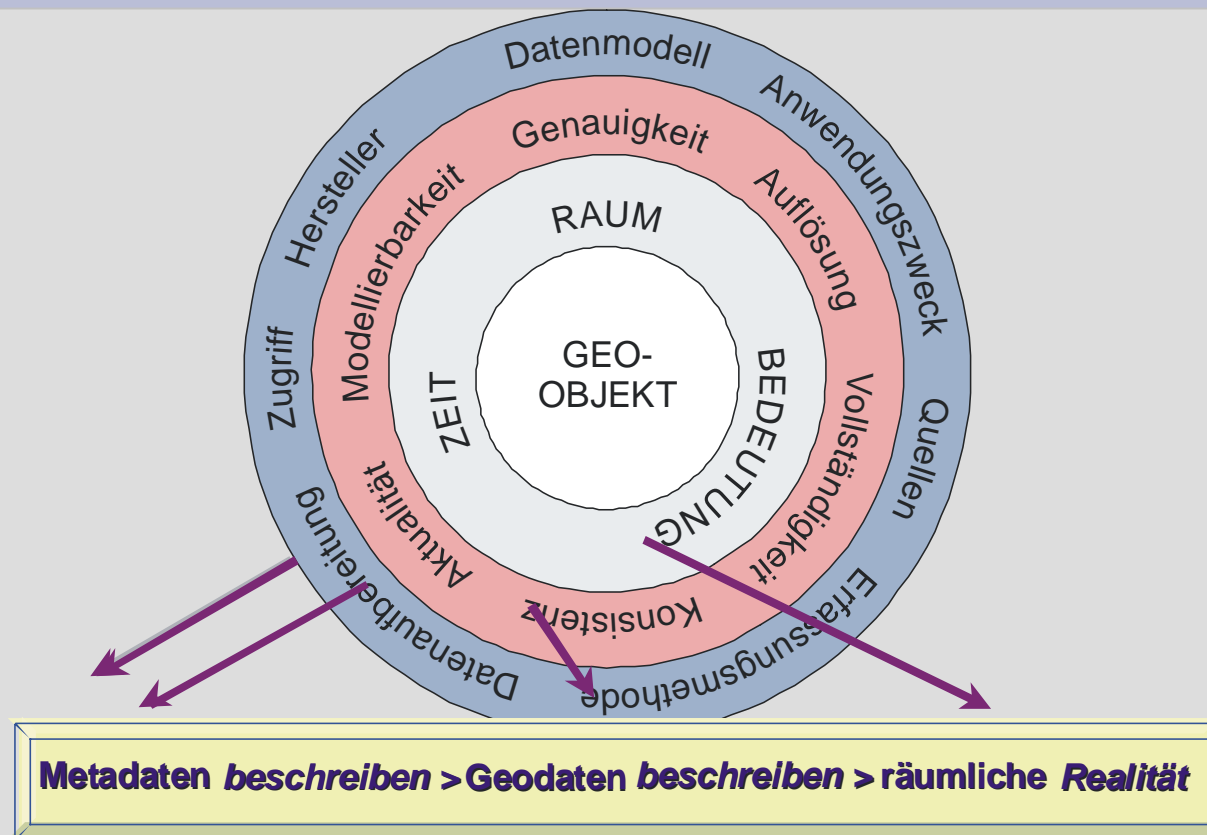
3) GeoForschungsZentrum Potsdam, Telegrafenberg A3, D-14478 Potsdam,
knb@gfz-potsdam.de



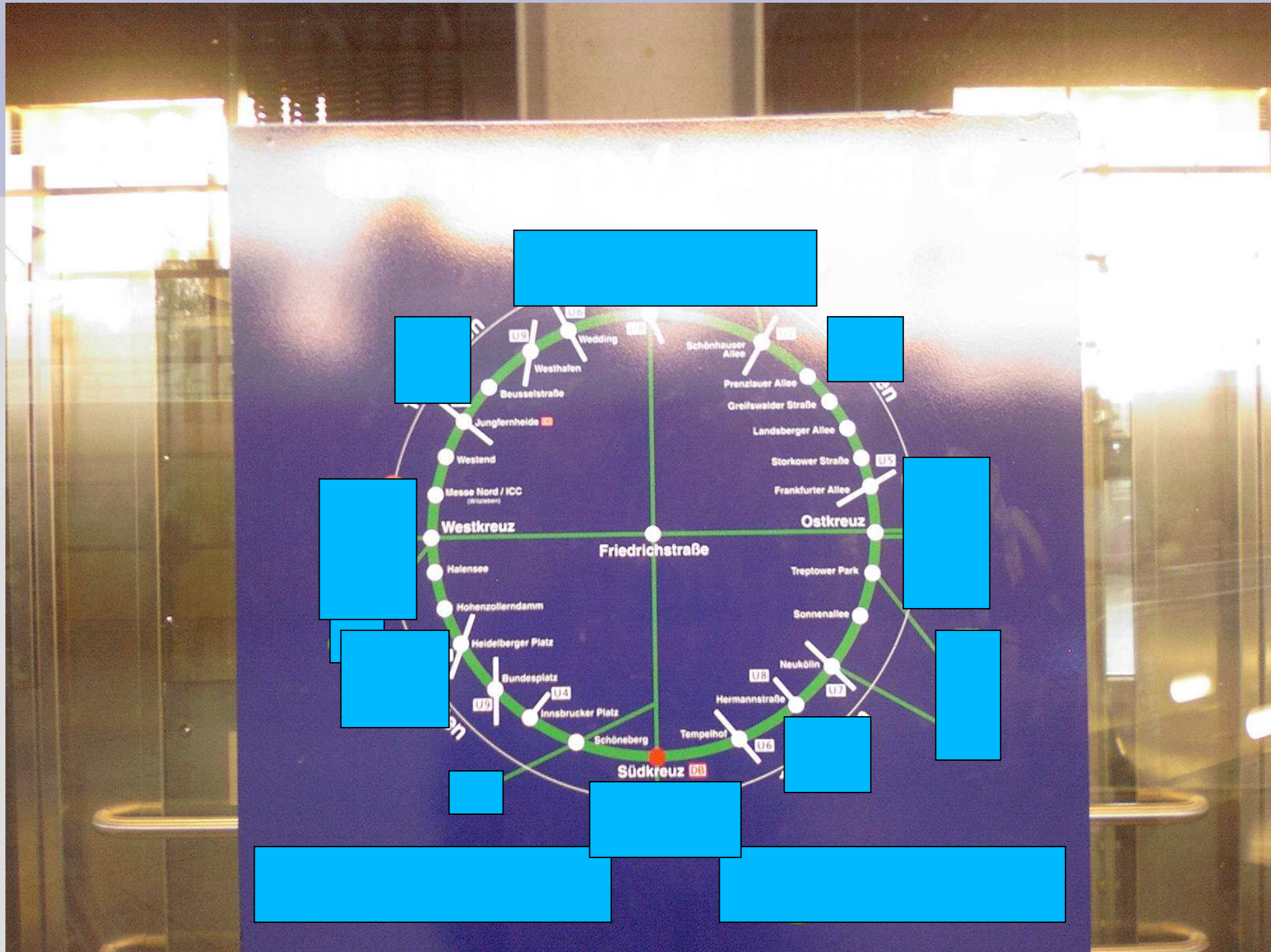
Übersicht

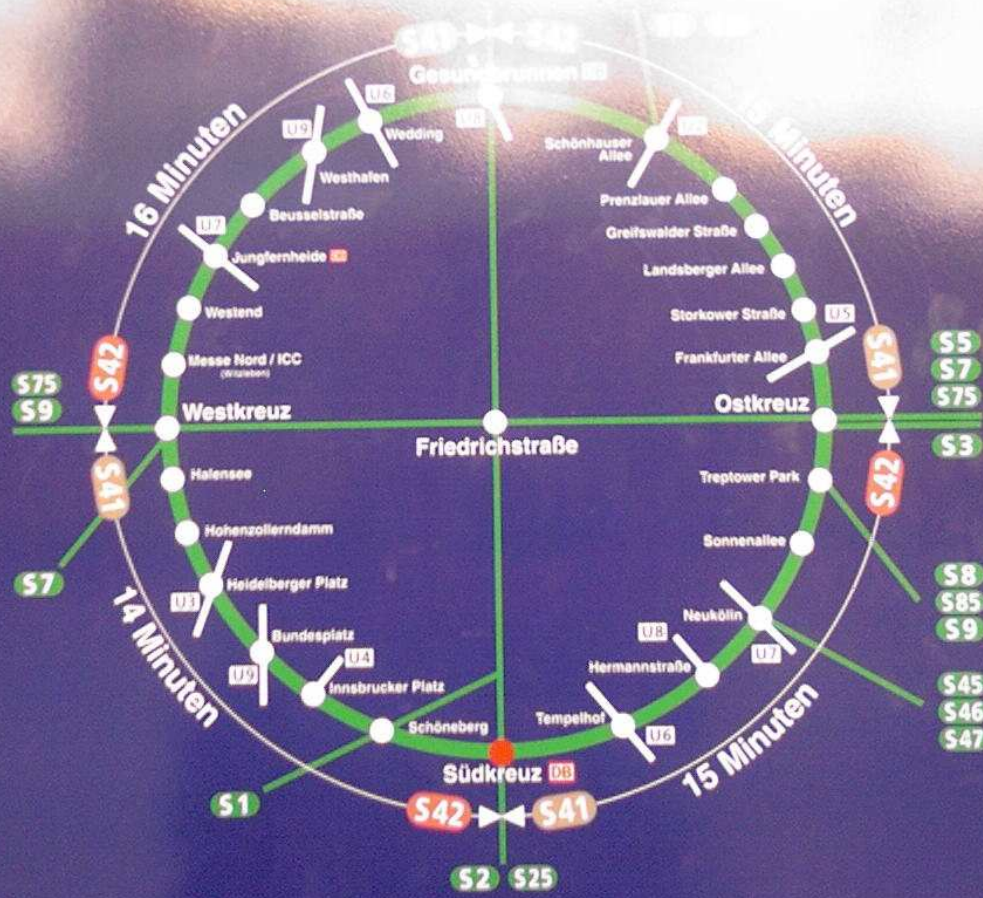
- Modularer Aufbau des Workshops:
- Theorie:
- Einführung / Standalone Interaktive Metadateneditoren (Wo)
- Libxml (St)
- UML (Kn)
- Praktische Übungen: “Wir basteln uns einen Metadateneditor / Vergleichen von Standalone Editoren

Metadaten



„Metadaten beschreiben räumliche Daten umfassend so, dass deren Eignung für einen bestimmten Anwendungszweck vorab beurteilt werden kann.“ (Strobl 1995).





● Eigener Standort



Geo-Metadatenstandards

- FGDC seit ca 1994
- Dublin Core (dct:spatial) seit ca 1998
- ISO 19115 seit ca 2003
- ... (andere)

FGDC: Federal Global Data Clearinghouse
(frühes Konzept Geodaten-Infrastruktur, USA)

ISO: International Standards Organisation

Relative Bedeutung der drei Metadaten-Standards

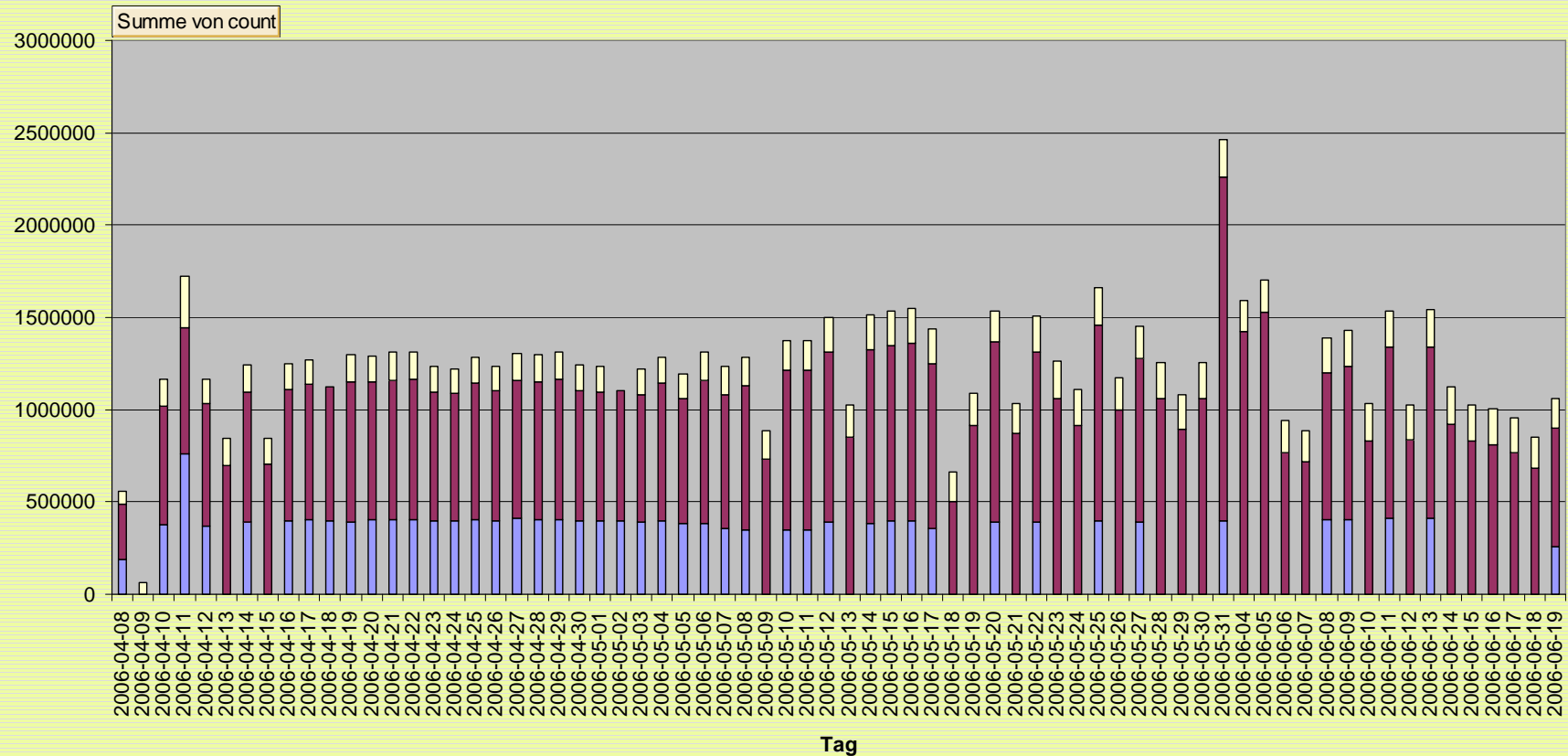
- ISO 19115, FGDC, DC:
 - Welcher Verbreitungsgrad?
 - Welche Metrik verwenden?
 - Webseiten-Anzahl, Wachstumsrate
 - Ermittelt anhand von Google Daterange Search
 - Beliebte Suchbegriffe
 - Google Trends

3 Monatstrend: Webseiten-Anzahl, Wachstumsrate (April 2006- Juni 2006)

Seitenfelder hierher ziehen

Geometradatenstandards als Schlüsselwort: Tägliche Aktualisierungen im Google Index

search term
▾
■ dublin core spatial metadata ■ FGDC spatial metadata □ ISO 19115 spatial metadata



Tag

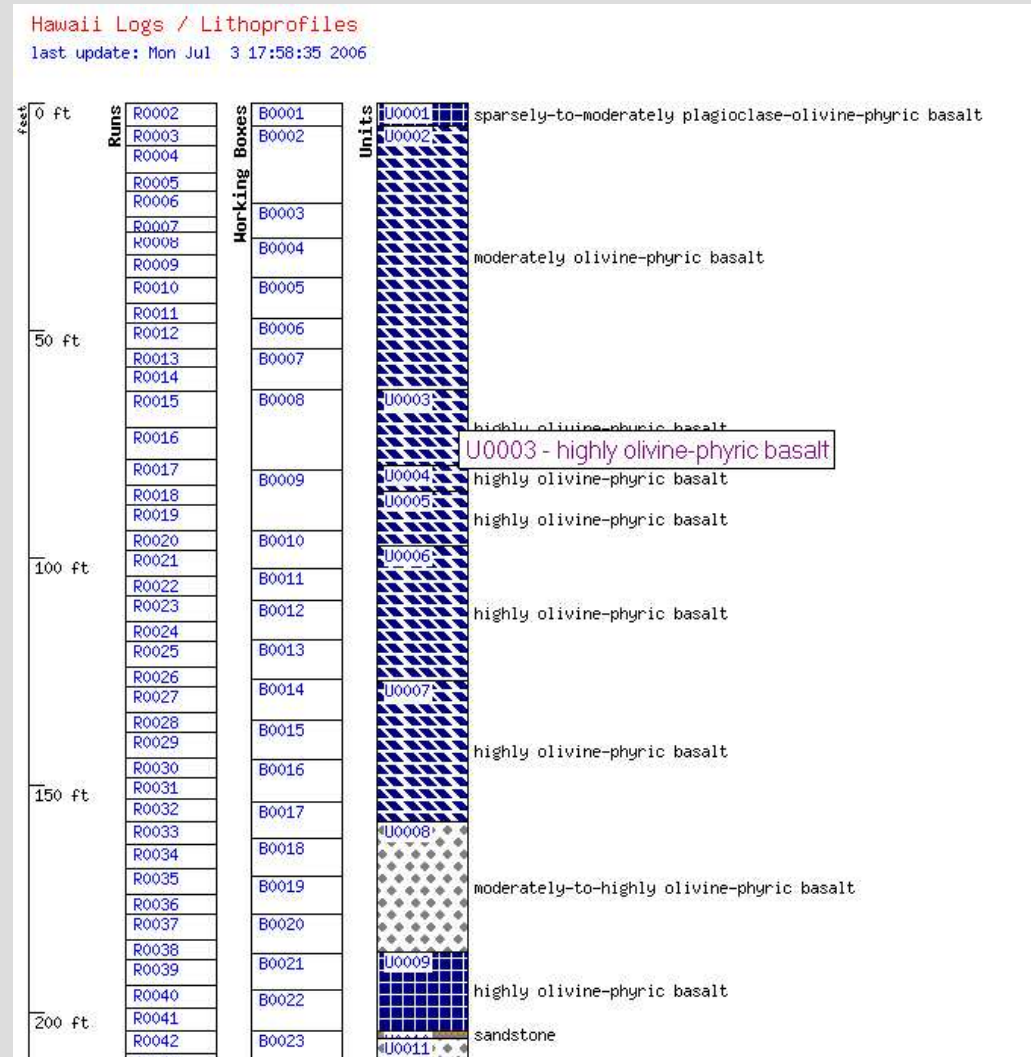
date ▾

Unsere Anforderungen an Editoren

- Allgemein
 - Aktuelle Versionen der Geodatenstandard(s) korrekt, vollständig unterstützen
 - Reich an Funktionalität, dennoch benutzerfreundlich
 - Flexible Ein- und Ausgabeformate
 - Unterstützung von Templates
 - Interaktives Editieren, und Zugriff per Script
 - Praxistauglichkeit:
 - 3 Testszenarien

Szenarios

- Szenario 1:
Eindimensionale Daten
 - Beschreibung einer Bohrung (1 x,y Koordinate plus „Schichtenverzeichnis“)
- Manuelle Bearbeitung
 - Interaktiver Editor
 - Leeres Metadaten-File erstellen, Daten eingeben

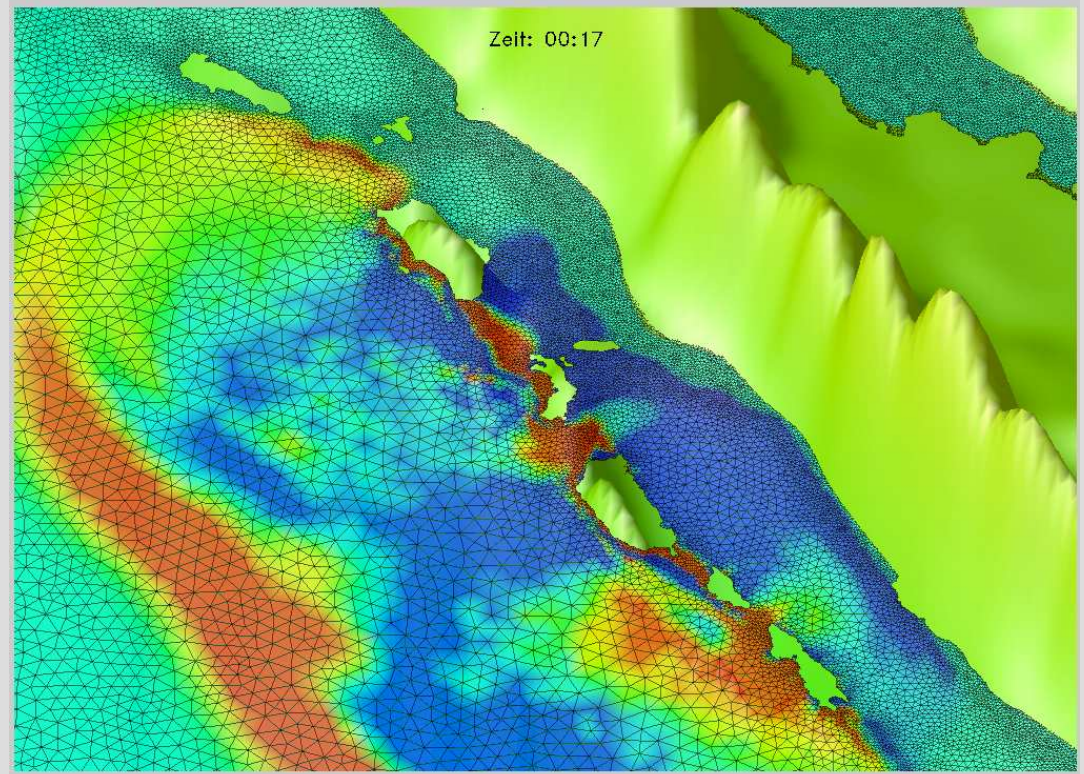


Szenario 2

- Flächige Geologische Karte
- Manuelle Bearbeitung der Metadaten
 - Hinzufügen/Aktualisieren von bereits existierendem, umfangreichen Fremd-Datensatz
 - Import/Export

Szenario 3

- 4-D Simulationen von Tsunamiwellen
- Erzeugung von Metadatenfiles für zahlreiche, parametrisierte Simulationen.
- Interaktives Template erzeugen, Zugriff per Script



Klassifikation der Editoren

- **Spezialeditoren**
 - Auf Geo-Metadatenstandard(s) zugeschnitten
- **Generische XML-Editoren**
 - Dienen allgemein der Bearbeitung von XML Daten
 - Flexibler, reicher an Funktionalität
- **(Webbasierte Angebote, Onlinedienste)**
 - nicht untersucht
 - z.B. Validierungsdienste, Konvertierungsdienste...

Ergebnisübersicht

- Siehe Artikel im Tagungsband
- Ausführlichere Darstellung der Merkmale auf Webseite
<http://dc110.gfz-potsdam.de/agit>

Editoren

Metadateneditor	Betriebs-System	Verfügbarkeit	Unterstützte Standards	Datenhaltung	Gruppe
ArcCatalog	Windows	kommerziell (Fa. ESRI)	FGDC	Proprietär-XML	Spezial
ArcView Metadata Extension	Unix, Windows	kommerziell (ArcView, Fa. ESRI)	FGDC	Proprietär-XML	Spezial
Authentic 2005	Windows	kommerziell (Fa. Altova)	W3C-XML-Schema	DB oder XML- (proprietär)	Generisch
CatMDEdit	Unix, Windows	frei	Dublin Core, FGDC, ISO		Spezial
Corpsmet95	Windows	eingestellt	FGDC		Spezial- EOL
Enraemed	Windows	frei	Dublin Core, ISO	SQL-Datenbank	Spezial-EOL
Infopath	Windows	kommerziell (Microsoft)	W3C-XML-Schema	DB oder XML- (proprietär)	Generisch
Mediator	Windows	frei	ISO-19115		Spezial-EOL
Preludio	Windows	gratis, komm. (Fa. Disy)	ISO-19115	Datenbank	Spezial
PubliStar	Linux, Windows	Kommz. (Fa. Delphi IMM)	ISO-19115	Datenbank (DB)	Spezial
USGS FGDC Metadata Tools	Unix, Windows	frei	FGDC	DB, Files	Spezial

EOL: End Of Life (Entwicklung und Vertrieb eingestellt)

Bewertung Editoren

- **Spezialeditoren**
 - sehr unterschiedliche Qualität, Extrembeispiele:
 - einfache Erfassungstools, z.T. Entwicklung eingestellt
 - komplexe Applikationen mit nahezu vollständiger Unterstützung von ISO Geodaten-Standard
 - oft nicht flexibel genug
- **Generische XML-Editoren**
 - Relativ hoher Aufwand für Konfiguration, Integration in Informationssysteme
 - Gangbare Alternative v.a. in größeren Projekten, für technikkompetentes Personal

Diskussion

- ISO 19115 Standard (seit 200x)
 - kleinerer Verbreitungsgrad, Anzahl Dokumente wächst
- FGDC, Dublin Core (seit 199x)
 - Größerer Verbreitungsgrad, Anzahl Docs \pm konstant
- Unterschiedliche Entwurfsprinzipien der Standards
Unterschiedliche Vorgehensmodelle bei der Standards-Entwicklung
 - prägt Leistungsumfang der Editoren
- Vielfalt der Spezialdisziplinen und Verwendungszwecke
 - Erfordert meist aufwendige Anpassung der Editorsoftware
 - „Extension Methodologies“ der Standards problematisch
- Schneller technischer und normativer Wandel

***Konzeptuelle Modellierung in
der ISO 19100er Serie von
Standards***

oder

***wie man ein Geodaten-
Metadaten-Profil
standardkonform selbst
entwickeln kann***

Knut Behrends

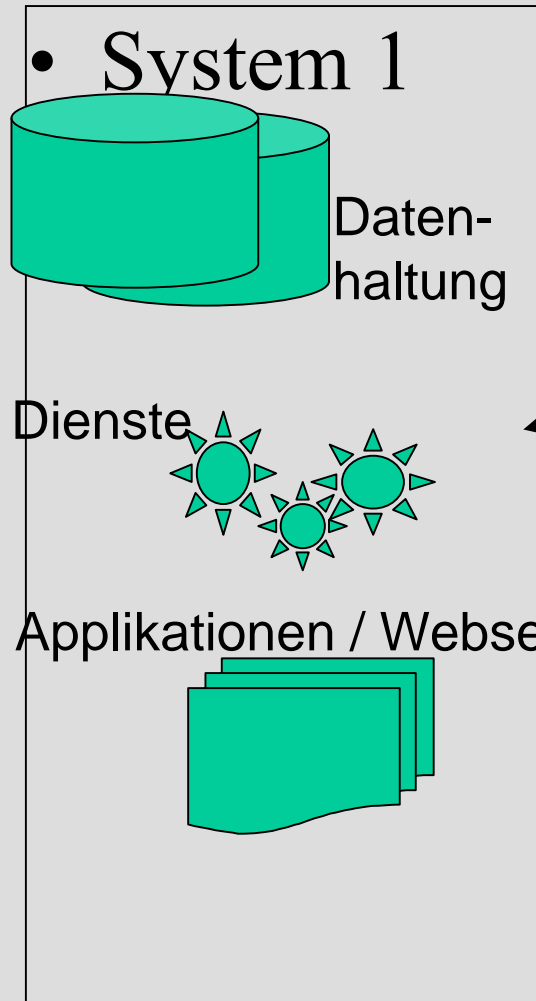
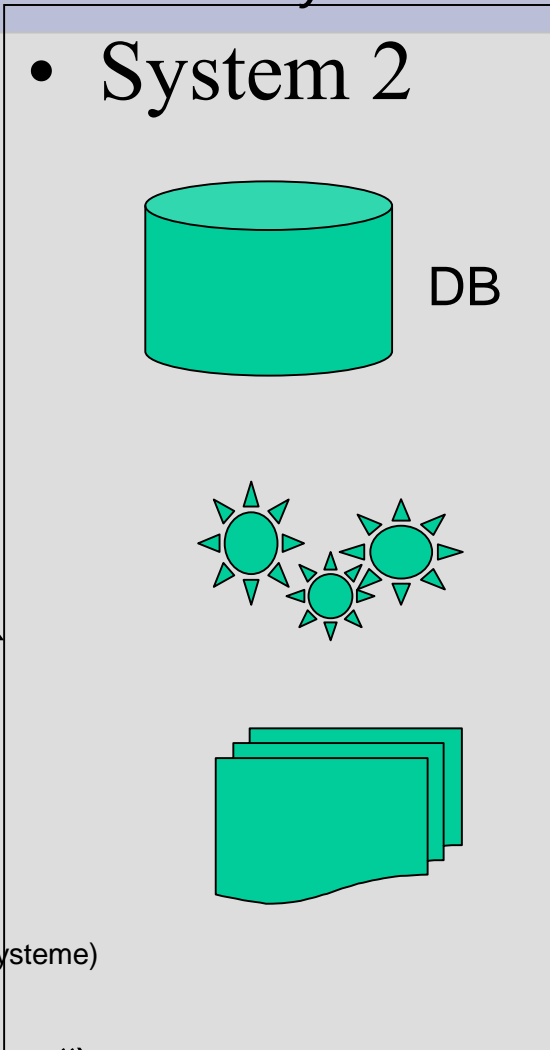
Gliederung

- Schnelldurchlauf
 - Konzeptuelle Modellierung in der ISO 191xx Familie von Standards
 - Domänen-Referenzmodell Geoinformation
 - Application schema

 - Umwandlungsregeln UML => XML => eigene Applikation
- Bezug zu Metadaten
 - Metadatenstandards 19115, 19139 in Grundzügen verstehen
 - Extension Mechanisms
- Praxisteil:
 - ISO 19115-Package „Data Lineage“: Prozessierungsschritte standardkonform dokumentieren
 - => d.h. Kleines Metadatenprofil entwickeln (standardkonform erweitern)
 - Metadateneditor selber herstellen aus XML Schema (per Drag & Drop mit Infopath oder Authentic)

Das Problem

Nicht nur in der Geo-Informatik, sondern allgemein bei Informationssystemen!



Normalerweise
Inkompatibel !

Lösung:
Interoperabel machen

Hilfsmittel:
Standards
(aber: entweder zu ungenau oder zu restriktiv!
Extreme Verschiedenheit der Fachgebiete und Systeme)

Abstraktionsprozess
(= „Konzeptuelle Modellierung“)

Keine UML Notation!

Abstraktionsprozess

- Abstraktionsprozess „Konzeptuelle Modellierung“ ebenfalls in Standards festgelegt und beschrieben
 - Siehe ISO 19100, 19101 (sehr kompakt)
 - Siehe
 - 10746 Informationssysteme
 - 14481 Open Distributed Processing (Verteiltes Rechnen)
- Die „Konzeptuelle Modellierung“ sollte nicht mehr als 4 „Meta-Ebenen“ haben
 - 4 Schichten sind genug!
 - Dies ist Erfahrungswissen/best practice
- Was sind denn nun Meta-Ebenen ?

Modellierung auf vier Meta Ebenen

Beispiel

Natürlichsprachliches Beispiel (Lexical Representation)

Strassen/Kreuzungen

Definition/Auswahl von Instrumenten für das Metamodell: z.B. „Wir folgen einer Modellbasis (Paradigma) wie z.B. Objektorientierung“

Meta-Meta-Modell-Ebene

„Wir benutzen Klassen/Objekte und Verknüpfungen (Assoziationen, Links,...)“

Meta-Modell-Ebene

„Es gibt Straßen und Kreuzungen. Eine Straße beginnt und endet an je 0 oder 1 Kreuzung.“

Modellebene

„Die Müllerstrasse mündet in die Turnstrasse und in die Pasteurstrasse“

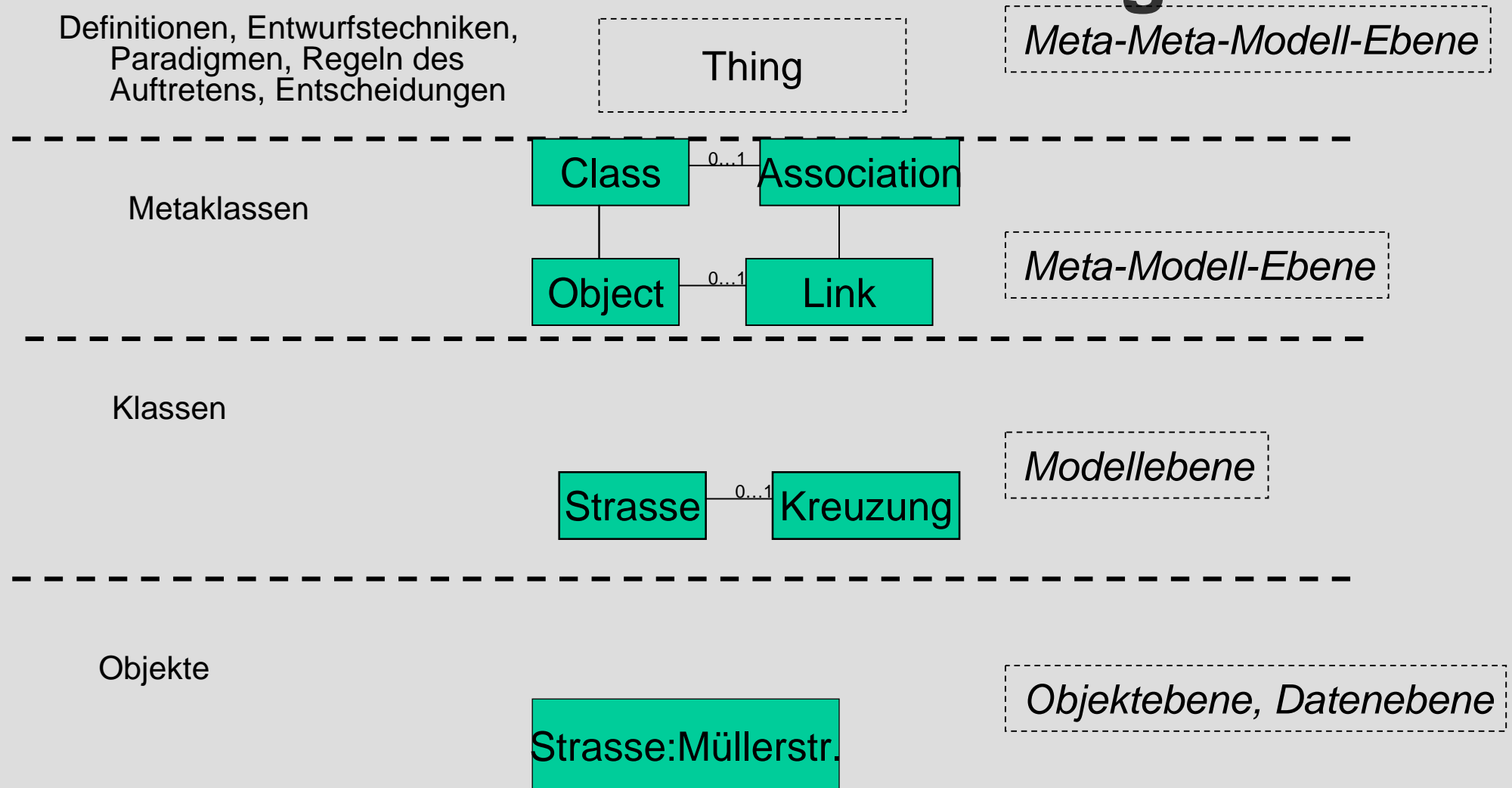
Objektebene, Datenebene

Modellierung auf vier Meta Ebenen

Beispiel

Graphische Repräsentation des Beispiels (*noch nicht in UML!*)

Strassen/Kreuzungen



Modellierung auf vier Meta Ebenen

Beispiel

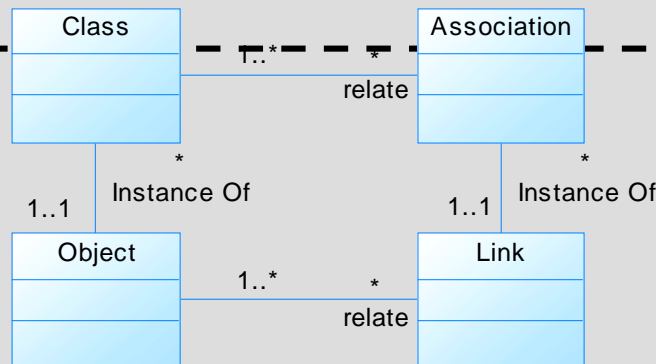
Graphische Representation des Beispiels (in UML!)

Strassen/Kreuzungen

Dinge, Entscheidungen,
Paradigmen, Regeln des
Auftretens

Meta-Meta-Modell-Ebene

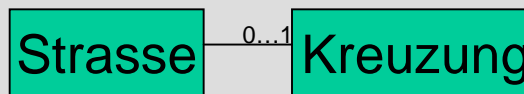
Metaklassen



Meta-Modell-Ebene

*Hinweis: für „wahres“ Metamodell, siehe Dokument
„UML Superstructure“, www.omg.org*

Klassen



*Modellebene, auch:
Application Model Level*

Objekte

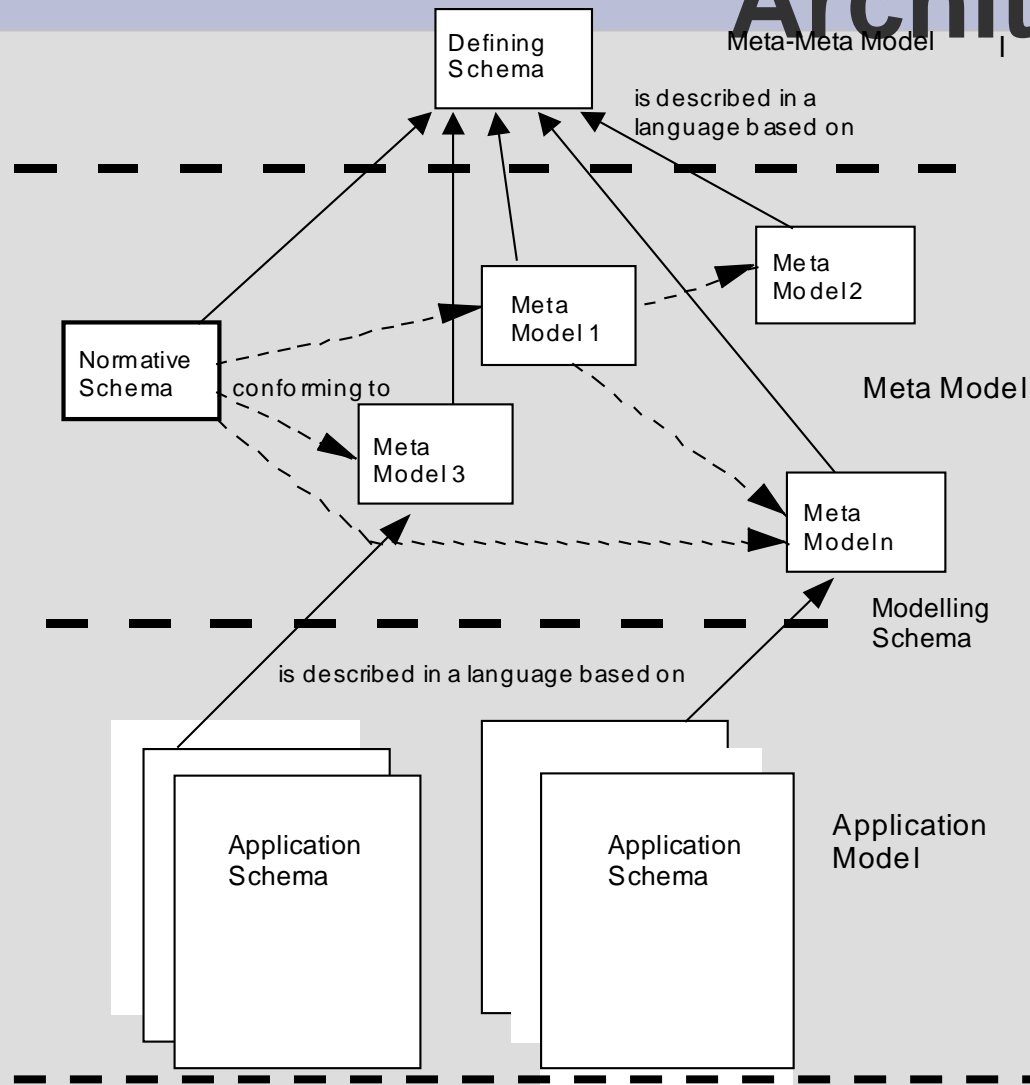
Strasse:Müllerstr.

Objektebene, Datenebene

ISO Conceptual Schema Modelling Facility

- Eine 4 - Schichten Modellierungsarchitektur
- Definiert v.a. in ISO 14481 (1990er Jahre)
- In ISO 191xx Standards ziemlich durchgängig verwendet
- Es gibt keine Standards für die Meta-Meta Ebene
- In 191xx Standards in vielen Diagrammen/Abbildungen
 - Unterste Ebene (Datenebene) weggelassen
 - Daher eigentlich nur 3 Ebenen sichtbar
 - Aber: Mehr Details auf Metamodell-Ebene und Application-Model-Ebene hinzugefügt

Conceptual Schema Modelling Facility (CSMF) Architektur



Meta-Meta-Model

Meta-Model

Application Model

Domänen-unabhängig. Nicht nur für Fachgebiet Geoinformation, sondern auch in ganz anderem Kontext benutzbar

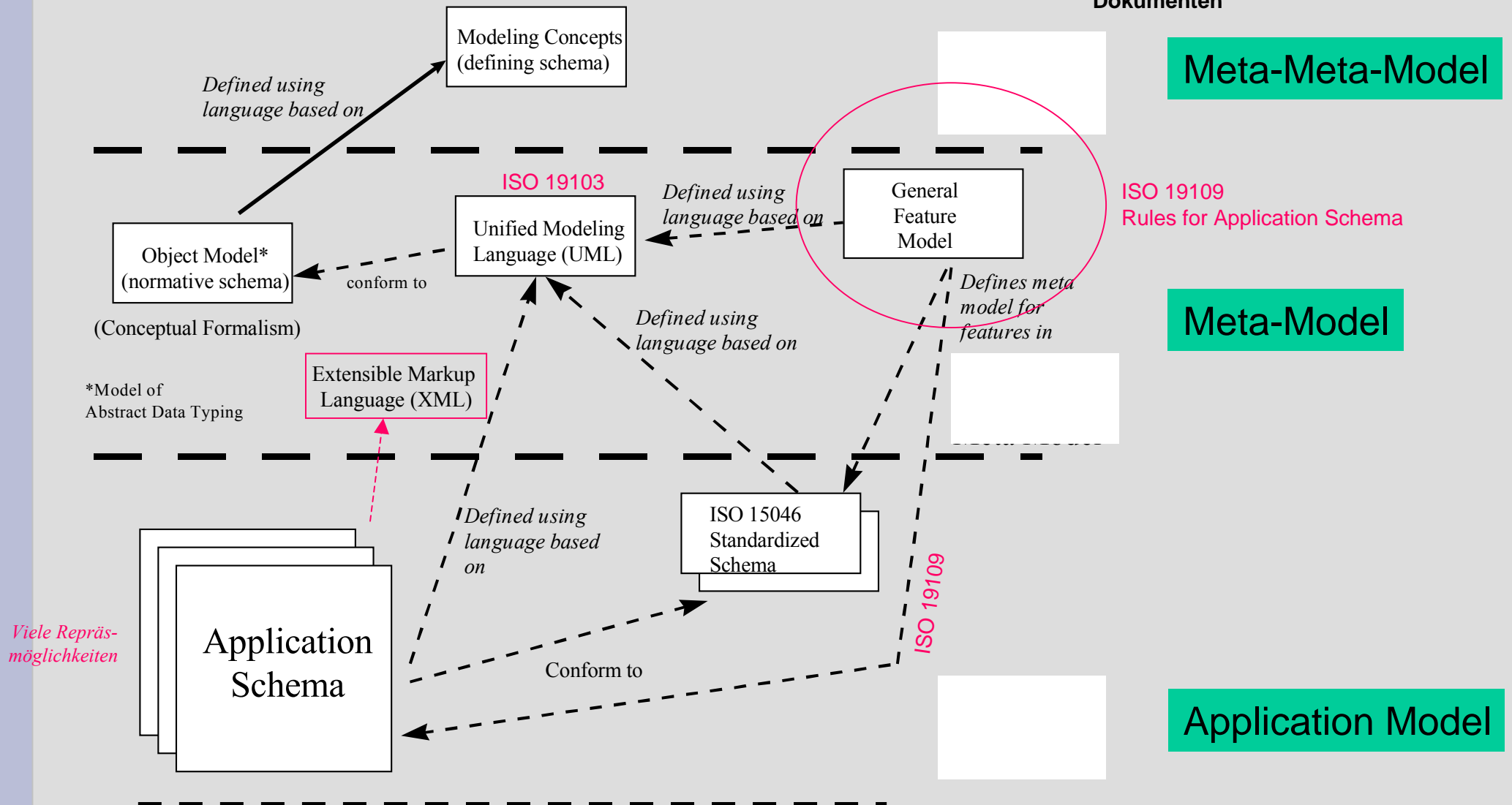
ISO 19100
ISO 14481

(Daten-Ebene spielt in CSMF keine Rolle)

Domänen-Referenzmodell

(„Domäne“ bezeichnet hier das Fachgebiet *Geoinformation*, modelliert in der CSMF)

Abbildung aus ISO 19100, 19101
Dokumenten



General Feature Model nach ISO 19109

- Metadaten sind eine Form von „Feature Attributes“. Alles weitere regelt ISO 19115

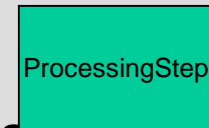
Definitionen

- **Metadata entity**

- set of metadata elements describing the same aspect of data
 - NOTE 1 May contain one or more metadata entities.
 - NOTE 2 Equivalent to a **class** in UML terminology.

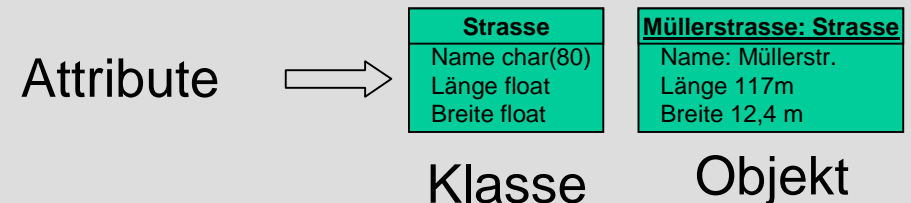
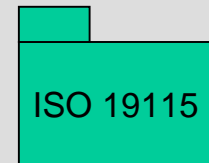
- **Metadata section**

- subset of metadata which consists of a collection of related metadata entities and metadata elements
- NOTE Equivalent to a **package** in UML terminology.



- **Metadata element**

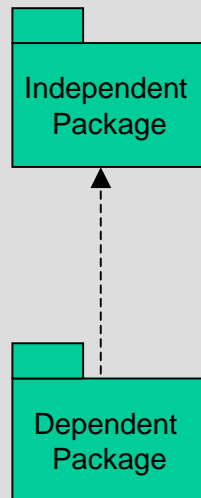
- discrete unit of **metadata** [ISO 19115]
 - NOTE 1 Metadata elements are unique within a metadata entity.
 - NOTE 2 Equivalent to an **attribute** in UML terminology.



Definitionen/Notationen

Dependency:

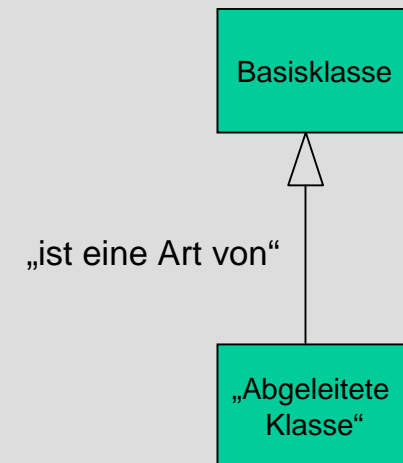
relationship between two modelling elements, in which a change to one modelling element (the independent element) will affect the other modelling element (the dependent element)



Aggregation: relationship

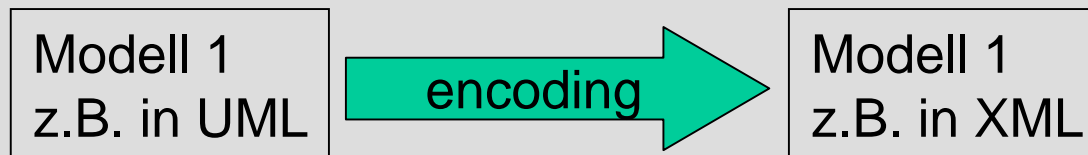


Generalization / IS-A Relation / Inheritance:



Definitionen

Encoding



Kritik an Metadatenstandards

- ISO 19115 geht davon aus dass sich Daten (D) und Metadatensätze (Md) trennen lassen
 - Das ist nicht immer der Fall: Beispiel „Wasserprobenverwaltung, Geschäftsprozesse „Untersuchen“ (U) und „Inventarisieren“ (I) :
 $D(U) = Md(I)$ und $Md(U) = D(I)$
- ISO Dokumente (pdf-Dateien)
 - Fehler: Rechtschreibung, z.T. UML-Modelle Aggregationsbeziehungen falsch herum
 - Inkonsistenzen in der Architektur
 - Operationen nicht spezifiziert, Attribute jedoch übergenau),
 - „Metadata Element“ = UML „Attribute“, etc.
 - Mehrfachvererbung wird benutzt (schwierig in XML umzusetzen)
 - Wichtige Diagramme liegen gerade nicht in UML vor, Einige Dokumente ständig überarbeitet, noch nicht fertig oder bereits wieder zurückgezogen (z.B. ISO 19102, Gesamtüberblick)
- UML entwickelt sich weiter
 - UML 1.3 wird Docs seit 2003 benutzt, veraltet. Aktuell ist UML 2.1
 - Inkonsistenter Gebrauch der UML-OCL (Object Constraint Language) “

Literaturhinweise

- **ISO TC211** <http://www.isotc211.org>
 - Webseiten mit Modellen zum Herunterladen, News, Dokumenten, etc.
- **Alhir, Sinan Si (1998): UML in a Nutshell. A desktop quick reference. O'Reilly. ISBN 1565924487**
 - Extrem analytisch werden viele Begriffe übergenau erklärt, z.T. redundant. Autor hebt immer wieder die Verwendung von UML als Mittel für Problemlösungen hervor. Hat aber auch jede Menge brauchbare Beispiele. Kleine Schrift, Großteil des Texts besteht aus „Bulleted Lists“. Veraltet da UML 1.1 beschrieben wird; aber trotzdem als Inspirationsquelle noch zu gebrauchen. Ein kuriose Buch.
- **Born, Marc; Holz, Eckhart; Kath, Olaf (2003): Softwareentwicklung mit der UML. Die neuen Entwurfstechniken UML 2, MOF 2 und MDA im Vergleich**
 - Gründliche und gut lesbare Einführung in UML 2. Dabei wird vom UML Metamodell ausgegangen, und die Konzepte und Sprachelemente Schritt für Schritt erklärt.
- **Michi Henning (2006): The Rise and Fall of CORBA.** <http://www.acm.org/publications/diagrams/Content/Content.aspx?cid=20060817> Date accessed Aug 17, 2006
 - Interessante, ausführliche Beschreibung warum „Design by Committee“ bei der CORBA Technologie unergiebig war und warum dieser Softwarestandard letztlich floppte. Dabei gilt CORBA noch als einer der erfolgreicheren Standards, da viele Industriegiganten vereint gegen Microsoft antraten. Der Autor geht auf Technische, Organisatorische, Politischen und Psychologische Missstände ein.
 - *A democratic process such as the OMG's is uniquely ill-suited for creating software. Despite the known procedural problems, however, the industry prefers to rely on large consortia to produce technology. Web services, the current silver bullet of middleware, uses a process much like the OMG's, and by many accounts, also suffers from infighting, fragmentation, lack of architectural coherence, design by committee, and feature bloat. It seems inevitable that Web services will enact a history quite similar to CORBA's.*
- **(anonym) (2005): Linus says No to ,Specs'.** <http://www slashdot.org/story.pl?cat=05/10/05/216237> Date accessed Aug 17, 2006
 - Meldung auf Technik-Nachrichten-Website **slashdot.org** zum Thema Spezifikationen (eigentlich Standards) von Hard- und Software. Ausgehend von Linus Thorvalds' Posting an Mailingliste der Linux-Kernel-Entwickler, gibt es eine erschöpfende Diskussion (mit 540 Beiträgen!) zum Thema Pro und Kontra von technischen Standards. (Standards sind entweder zu restriktiv formuliert oder zu ungenau.)
- **Andrea Trinkwalder (2006) Für die Ewigkeit, Metadatenstandards für Bildarchiv.** c't 2006, Heft 16, Seite 156-158
 - Interessanter Artikel über Metadaten in der Digitalfotografie. Im Header von Digitalfotos können auch technische Metadaten untergebracht werden. Leider gibt es derzeit drei konkurrierende Standards, und kaum ein Softwaretool unterstützt alle davon. Folge ist, dass beim Nachbearbeiten der Metadaten in den verschiedenen Dateihedern schnell Informationen verloren gehen. Darüber hinaus sind die Erweiterungsmechanismen schlecht definiert oder sogar inkompatibel.
- **Object Management Group (2006): Catalog of OMG Specifications.** http://www.omg.org/technology/documents/spec_catalog.htm
 - Hier gibt es frei erhältlich, zum Herunterladen:
 - UML Superstructure: Das Meta-Modell der UML. Als PDF Datei , 700 S.
 - UML Infrastructure : Das Meta-Meta-Modell der UML als PDF Datei
 - UML Infrastructure und UML Superstructure als Rational Rose Model (Diagramme und Klassendefinitionen)

Weitere Folien

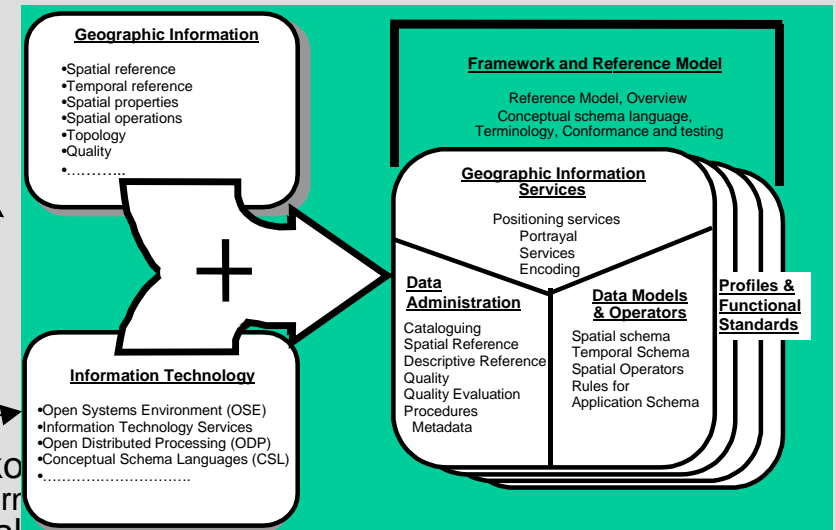
- ... falls niemand Fragen stellt

Geoinformationstandards integrieren...

- Geoinformations-Fachwissen
 - z.B. Projektionen, Metadaten...

...und...

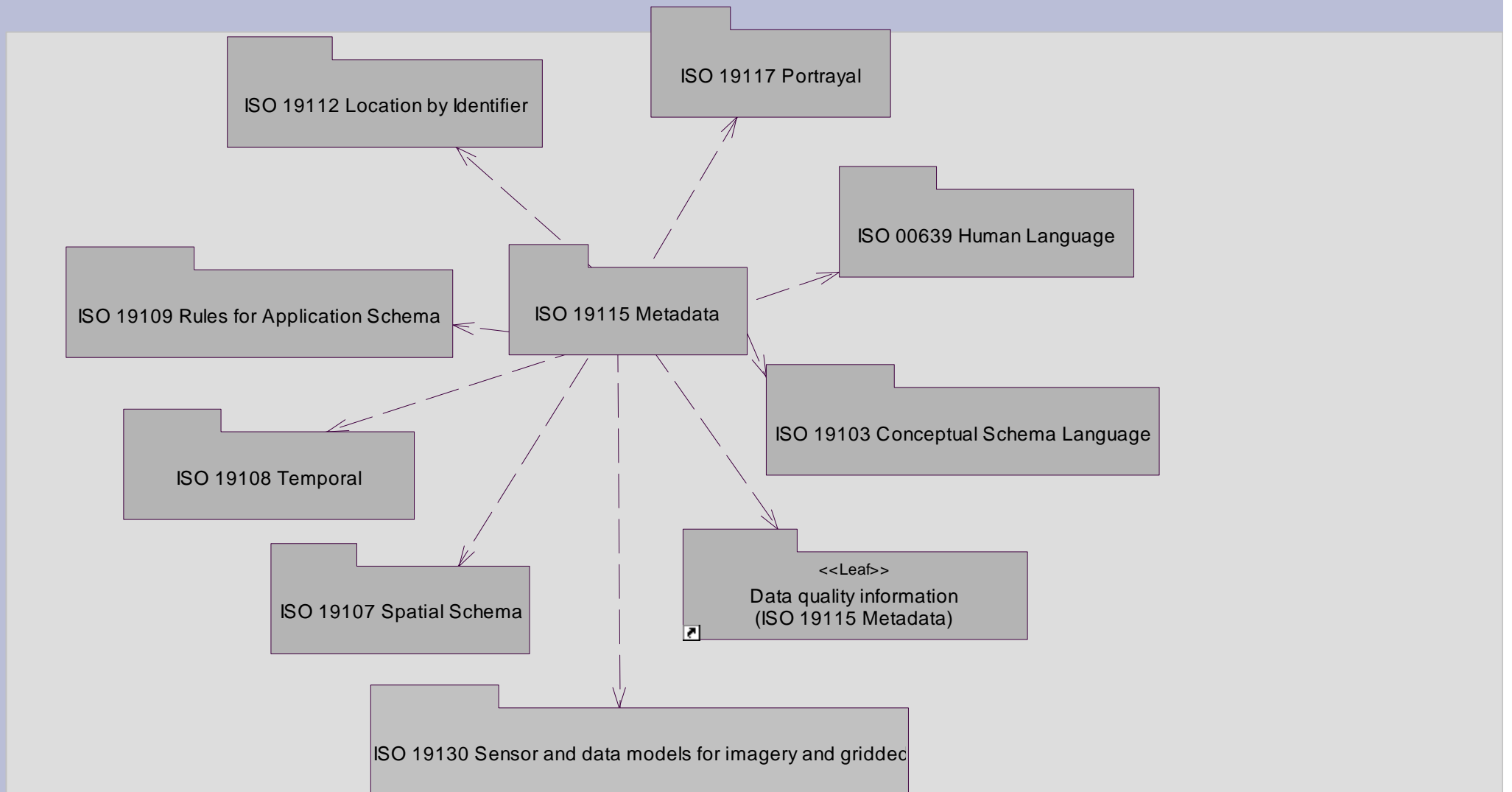
- Informatik-Fachwissen
 - bereits vorhandene ISO Standards wiederverwenden, kombinieren
 - Zeit- u. Datumswerte, Zeichensätze, Maßeinheiten, Länderkürzel
 - Entwurfstechnik „Konzeptuelle Modellierung“: ein Abstraktionsprozess
 - Objektorientierter Absatz
 - UML benutzen (Industriestandard, noch kein fertiger ISO Standard, wird ISO 15xxx)
 - Andere Prinzipien
 - 100% Prinzip (Vollständigkeit, alle relevanten Teile sollten erfasst werden)
 - Helsinki Prinzip... siehe ISO 19101



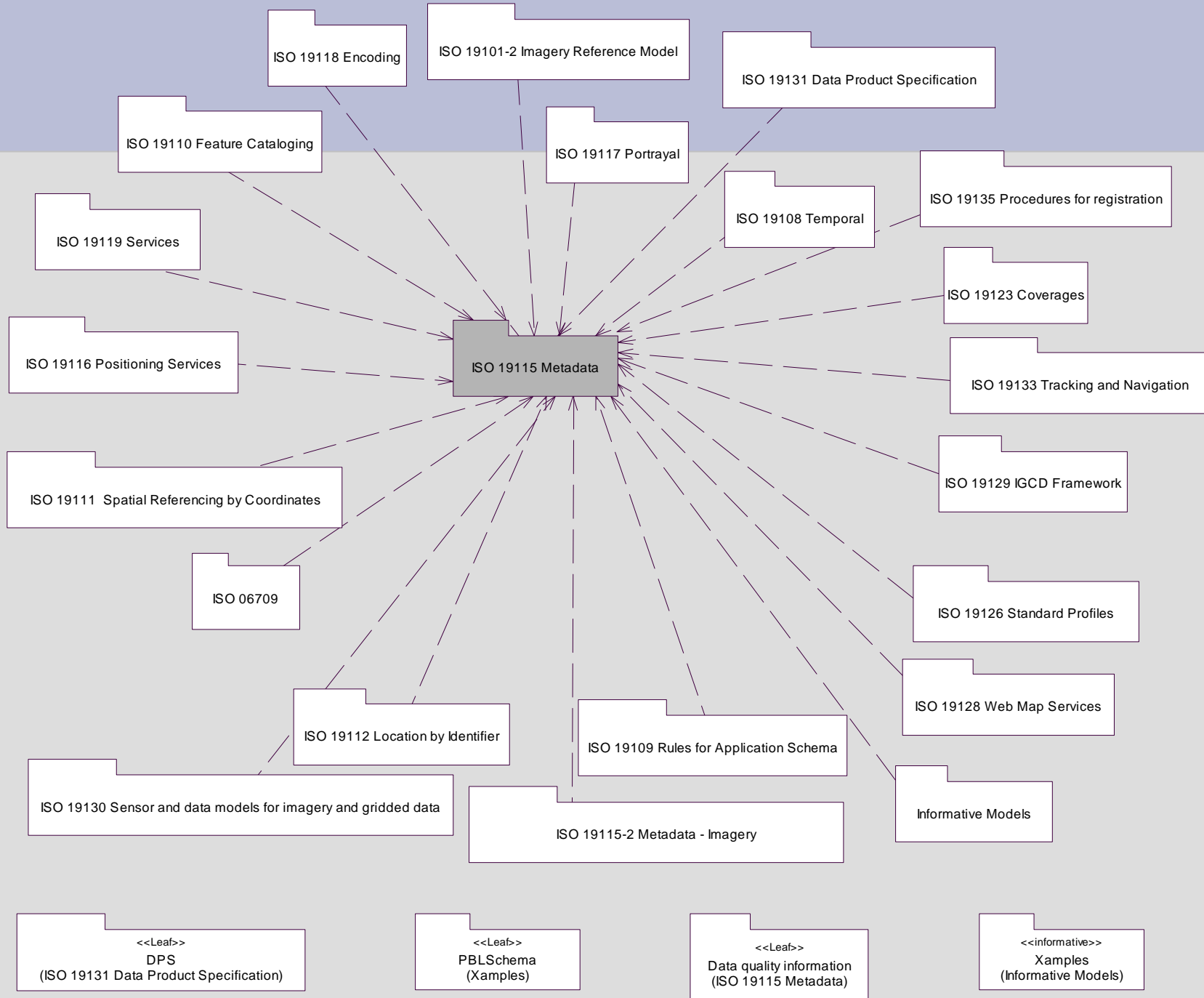
ISO 19100 Serie von Standards für Geoinformation

- ISO 19100: Übersicht
- ISO 19101: Domänenreferenz-Modell
 - 4-Schichtenarchitektur für Geoinformation
- ISO 19103: Spezif. d. benötigten Untermenge der UML, spezielle Konstrukte (z.B. wichtige Datentypen für Geoinformation), Requirements, Namenskonventionen ...
- ISO 19109: Regeln für *Aufstellung* von „Application Schemas“ (= vorzugsweise UML-Modellen)
- ISO 19118: Regeln zum *Austausch* von UML-Modellen, Umwandlungsregeln zum Ablegen in Dateien (XML Files)
- ISO 19115: Metadaten (abstrakt/konzeptuell)
- ISO 19139: Metadaten (in XML umgesetzt)

Supplier Packages für ISO 19115 (ISO 19115 ist abhängig von diesen Standards)

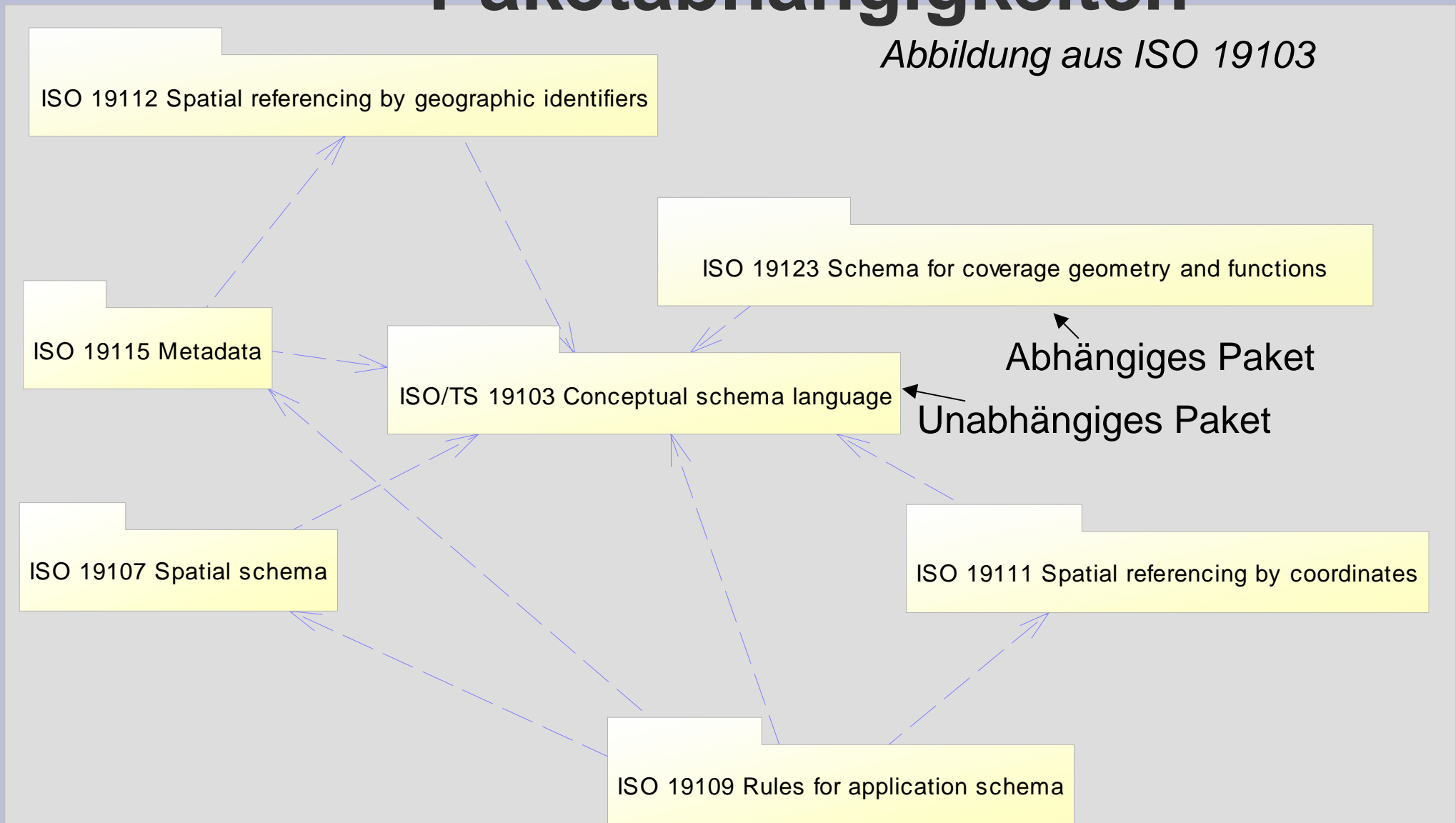


Client Packages von ISO 19115-Metadaten (Diese Standards sind von ISO 19115 abhängig)



Kleiner Ausschnitt aus dem Netzwerk der Paketabhängigkeiten

Abbildung aus ISO 19103



UML in ISO 191xx

UML

- Standardisierte Notation
- Paradigma: Objektorientierte Analyse und Design

Für ISO 191xx

- Version: 1.3, 1.4,
- Brauche nur „statische“ Teile
 - D.h. nur Strukturdiagramm-Notation (Klassen, Packages, Associations, etc)
 - Keine dynamischen Ablaufdiagramme, etc.

Modellierung auf vier Meta Ebenen

Beispiel

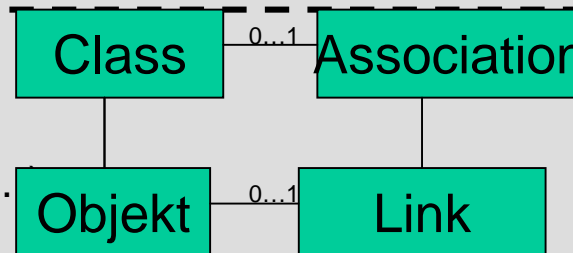
Hier alles in 1 diagramm – z.T. überfrachtet

Definition/Auswahl von Instrumenten für das Metamodell: z.B. „Wir folgen einer Modellbasis (Paradigma), z.B. Objektorientiert

Strassen/Kreuzungen

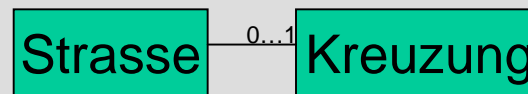
Meta-Meta-Modell-Ebene

Klasse
Verknüpfungen
(Assoziation, Relation, ...)



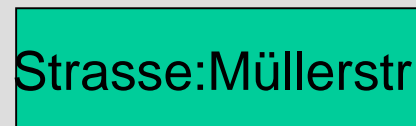
Meta-Modell-Ebene

Es gibt Straßen und Kreuzungen. Eine Straße beginnt und endet an 0 oder 1 Kreuzung.



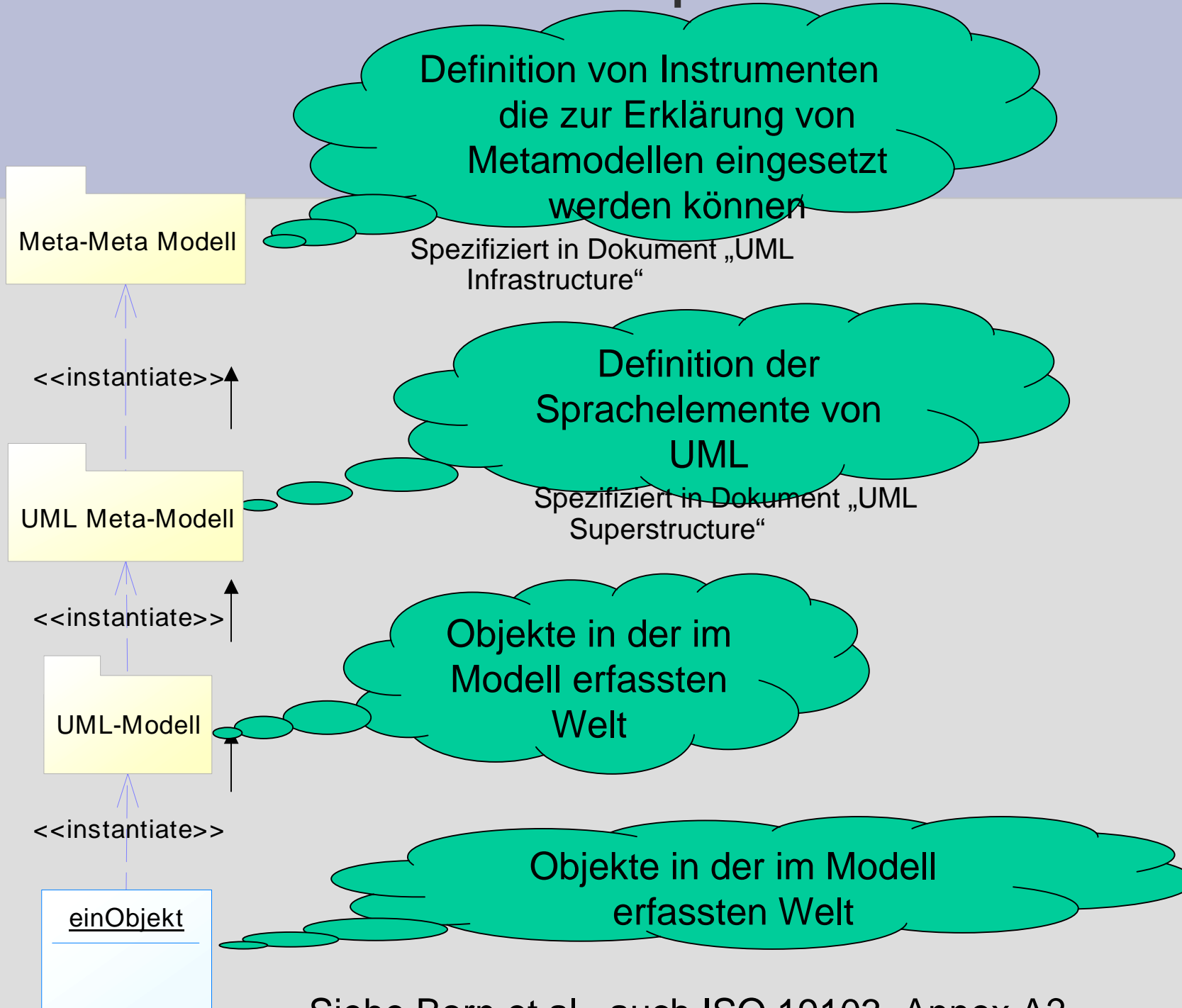
Modellebene

„Die Müllerstrasse mündet in die Turnstrasse und in die Pasteurstrasse“



Objektebene, Datenebene

Sprache UML selbst auch in 4 Schichten Architektur spezifiziert



Siehe Born et al., auch ISO 19103, Annex A?

ISO/TC 211 Projects (1)

Color legend: Draft International Standard - DIS,
Final Draft International Standard - FDIS, International Standard - IS

Stand: 05/2006

- ISO 6709:1983 - Standard representation of latitude, longitude and altitude for geographic point locations
- ISO 6709 - Revision
- ISO 19101 - Reference model
- ISO 19101-2 – Reference Model – Part 2: Imagery
- ISO/TS 19103 - Conceptual schema language
- ISO 19104 - Terminology
- ISO 19105 - Conformance and testing
- ISO 19106 - Profiles
- ISO 19107 - Spatial schema
- ISO 19108 - Temporal schema
- ISO 19109 - Rules for application schema
- ISO 19110 - Feature cataloguing methodology
- ISO 19111 - Spatial referencing by coordinates
- ISO 19111 - Revision
- ISO 19112 - Spatial referencing by geographic identifiers
- ISO 19113 - Quality principles
- ISO 19114 - Quality evaluation procedures
- ISO 19115 – Metadata
- ISO 19115-2 – Metadata – Part 2: Extensions for imagery and gridded data
- ISO 19116 - Positioning services
- ISO 19117 – Portrayal
- ISO 19118 - Encoding
- ISO 19119 - Services
- ISO/TR 19120 - Functional standards
- ISO/TR 19121 - Imagery and gridded data

ISO/TC 211 Projects (2)

Color legend: Draft International Standard - DIS,
Final Draft International Standard - FDIS, International Standard - IS

Stand: 05/2006

- ISO/TR 19122 - Qualifications and certification of personnel
- ISO 19123 - Schema for coverage geometry and functions
- ISO/RS 19124 - Imagery and gridded data components
- ISO 19125 - Simple feature access – Part 1-2
- ISO 19127 - Geodetic codes and parameters
- ISO 19128 - Web Map Server Interface
- ISO 19129 - Imagery, gridded and coverage data framework
- ISO 19130 - Sensor and data model for imagery and gridded data
- ISO 19131 - Data product specification
- ISO 19132 - Location based services possible standards
- ISO 19133 - Location based services tracking and navigation
- ISO 19134 - Multimodal location based services for routing and navigation
- ISO 19135 - Procedures for item registration
- ISO 19136 - Geography Markup Language (GML)
- ISO 19137 - Generally used profiles of the spatial schema and of similar important other schemas
- ISO 19138 - Data quality measures
- ISO 19139 - Metadata – Implementation specification
- ISO 19141 - Schema for moving features
- ISO 19142 - Web Feature Service
- ISO 19143 - Filter encoding
- Stage 0 - Am. to ISO 19113:2002 - Quality principles and ISO 19115:2003 - Metadata



The XML C parser and toolkit of Gnome

libxml

Main Menu

- [Home](#)
- [Reference Manual](#)
- [Introduction](#)
- [FAQ](#)
- [Developer Menu](#)
- [Reporting bugs and getting help](#)
- [How to help](#)
- [Downloads](#)
- [Releases](#)
- [XML](#)
- [XSLT](#)
- [Validation & DTDs](#)
- [Encodings support](#)
- [Catalog support](#)
- [Namespaces](#)
- [Contributions](#)
- [Code Examples](#)
- [API Menu](#)
- [XML Guidelines](#)
- [Recent Changes](#)

Related links

- [Mail archive](#)
- [XSLT libxslt](#)
- [DOM gdome2](#)
- [XML-DSig xmlsec](#)
- [FTP](#)
- [Windows binaries](#)
- [Solaris binaries](#)
- [MacOsX binaries](#)
- [C++ bindings](#)

"Programming with libxml2 is like the thrilling embrace of an exotic stranger." Mark Pilgrim

Libxml2 is the XML C parser and toolkit developed for the Gnome project (but usable outside of the Gnome platform), it is free software available under the [MIT License](#). XML itself is a metalanguage to design markup languages, i.e. text language where semantic and structure are added to the content using extra "markup" information enclosed between angle brackets. HTML is the most well-known markup language. Though the library is written in C [a variety of language bindings](#) make it available in other environments.

Libxml2 is known to be very portable, the library should build and work without serious troubles on a variety of systems (Linux, Unix, Windows, CygWin, MacOS, MacOS X, RISC Os, OS/2, VMS, QNX, MVS, ...)

Libxml2 implements a number of existing standards related to markup languages:

- the XML standard: <http://www.w3.org/TR/REC-xml>
- Namespaces in XML: <http://www.w3.org/TR/REC-xml-names/>
- XML Base: <http://www.w3.org/TR/xmlbase/>
- [RFC 2396](http://www.ietf.org/rfc/rfc2396.txt) : Uniform Resource Identifiers <http://www.ietf.org/rfc/rfc2396.txt>
- XML Path Language (XPath) 1.0: <http://www.w3.org/TR/xpath>
- HTML4 parser: <http://www.w3.org/TR/html401/>
- XML Pointer Language (XPointer) Version 1.0: <http://www.w3.org/TR/xptr>
- XML Inclusions (XInclude) Version 1.0: <http://www.w3.org/TR/xinclude/>
- ISO-8859-x encodings, as well as [rfc2044](#) [UTF-8] and [rfc2781](#) [UTF-16] Unicode encodings, and more if using iconv support
- part of SGML Open Technical Resolution TR9401:1997
- XML Catalogs Working Draft 06 August 2001: <http://www.oasis-open.org/committees/entity/spec-2001-08-06.html>
- Canonical XML Version 1.0: <http://www.w3.org/TR/xml-c14n> and the Exclusive XML Canonicalization CR draft <http://www.w3.org/TR/xml-exc-c14n>
- Relax NG, ISO/IEC 19757-2:2003, <http://www.oasis-open.org/committees/relax-ng/spec-20011203.html>
- W3C XML Schemas Part 2: Datatypes REC 02 May 2001
- W3C [xml:id](#) Working Draft 7 April 2004

In most cases libxml2 tries to implement the specifications in a relatively strictly compliant way. As of release 2.4.16, libxml2 passed all 1800+ tests from the [OASIS XML Tests Suite](#).

To some extent libxml2 provides support for the following additional specifications but doesn't claim to implement them completely:

- Document Object Model (DOM) <http://www.w3.org/TR/DOM-Level-2-Core/> the document model, but it doesn't implement the API itself, [gdome2](#) does this on top of libxml2

Liste der Schnittstellen (APIs) der libxml2

Reference Manual for libxml2 - Mozilla Firefox

API Menu

Search ...

- [Main Menu](#)
- [Developer Menu](#)
- [Code Examples](#)
- [API Menu](#)
- [Parser API](#)
- [Tree API](#)
- [Reader API](#)
- [XML Guidelines](#)
- [ChangeLog](#)

API Indexes

- [Alphabetic](#)
- [Constructors](#)
- [Functions/Types](#)
- [Modules](#)
- [Symbols](#)

Related links

- [Mail archive](#)
- [XSLT libxslt](#)
- [DOM gdome2](#)
- [XML-DSig xmlsec](#)
- [FTP](#)
- [Windows binaries](#)
- [Solaris binaries](#)
- [MacOsX binaries](#)
- [C++ bindings](#)
- [PHP bindings](#)
- [Pascal bindings](#)
- [Ruby bindings](#)
- [Tcl bindings](#)
- [Bug Tracker](#)

Table of Contents

- [DOCBparser](#): old DocBook SGML parser
- [HTMLparser](#): interface for an HTML 4.0 non-verifying parser
- [HTMLtree](#): specific APIs to process HTML tree, especially serialization
- [SAX](#): Old SAX version 1 handler, deprecated
- [SAX2](#): SAX2 parser interface used to build the DOM tree
- [c14n](#): Provide Canonical XML and Exclusive XML Canonicalization
- [catalog](#): interfaces to the Catalog handling system
- [chvalid](#): Unicode character range checking
- [debugXML](#): Tree debugging APIs
- [dict](#): string dictionary
- [encoding](#): interface for the encoding conversion functions
- [entities](#): interface for the XML entities handling
- [globals](#): interface for all global variables of the library
- [hash](#): Chained hash tables
- [list](#): lists interfaces
- [nanoftp](#): minimal FTP implementation
- [nanohhttp](#): minimal HTTP implementation
- [parser](#): the core parser module
- [parserInternals](#): internals routines exported by the parser.
- [pattern](#): pattern expression handling
- [relaxng](#): implementation of the Relax-NG validation
- [schemasInternals](#): internal interfaces for XML Schemas
- [schematron](#): XML Schemastron implementation
- [threads](#): interfaces for thread handling
- [tree](#): interfaces for tree manipulation
- [uri](#): library of generic URI related routines
- [valid](#): The DTD validation
- [xinclude](#): implementation of XInclude
- [xlink](#): unfinished XLink detection module
- [xmlIO](#): interface for the I/O interfaces used by the parser
- [xmlautomata](#): API to build regexp automata
- [xmlerror](#): error handling
- [xmlexport](#): macros for marking symbols as exportable/importable.
- [xmlmemory](#): interface for the memory allocator
- [xmlmodule](#): dynamic module loading
- [xmlreader](#): the XMLReader implementation
- [xmlregexp](#): regular expressions handling
- [xmlsave](#): the XML document serializer
- [xmlschemas](#): incomplete XML Schemas structure implementation
- [xmlschematypes](#): implementation of XML Schema Datatypes
- [xmlstring](#): set of routines to process strings
- [xmlunicode](#): Unicode character APIs
- [xmlversion](#): compile-time version informations
- [xmlwriter](#): text writing API for XML
- [xpath](#): XML Path Language implementation
- [xpathInternals](#): internal interfaces for XML Path Language implementation
- [xpointer](#): API to handle XML Pointers

Fertig

Beispiel eines einfachen C-Programms zum Schreiben einer XML-Datei

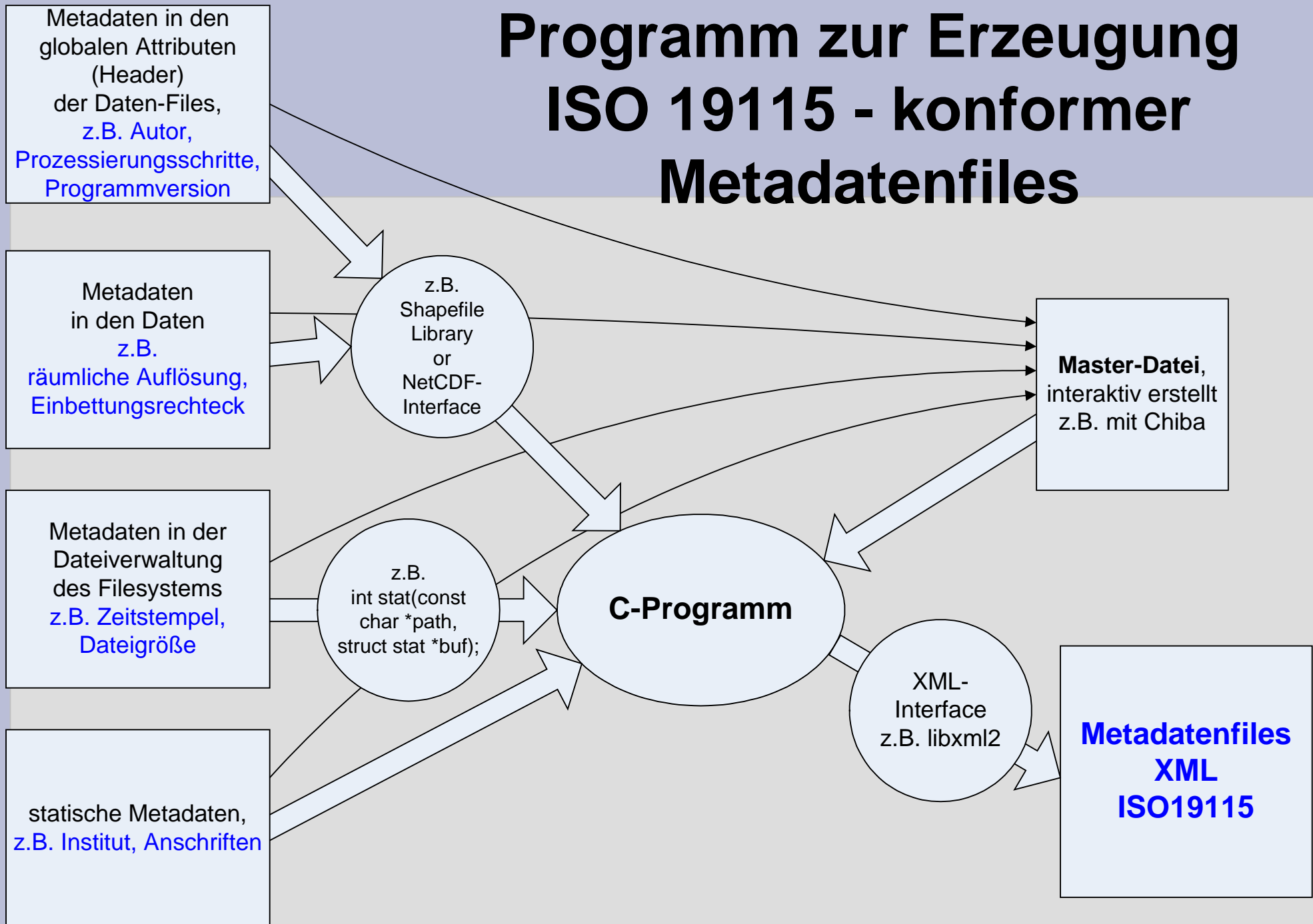
```
#include <libxml/xmlwriter.h>
#include <unistd.h>

int main ()
{ xmlDocPtr doc;
  xmlNodePtr root_node, ident, resparty, dataidinfo, citation, idcitation;

  doc = xmlNewDoc (BAD_CAST "1.0");
  root_node = xmlNewNode (NULL, BAD_CAST "MD_Metadata");
  xmlDocSetRootElement (doc, root_node);
  dataidinfo = xmlNewChild (root_node, NULL, BAD_CAST "identificationInfo", BAD_CAST NULL);
  ident = xmlNewChild (dataidinfo, NULL, BAD_CAST "MD_DataIdentification", BAD_CAST NULL);
  idcitation = xmlNewChild (ident, NULL, BAD_CAST "citation", BAD_CAST NULL);
  citation = xmlNewChild (idcitation, NULL, BAD_CAST "CI_Citation", BAD_CAST NULL);
  xmlNewChild (citation, NULL, BAD_CAST "title", BAD_CAST "Tsunami Simulation");
  resparty = xmlNewChild (citation, NULL, BAD_CAST "CI_ResponsibleParty", BAD_CAST NULL);
  xmlNewChild (resparty, NULL, BAD_CAST "individualName",
    xmlEncodeEntitiesReentrant (NULL, BAD_CAST "PD. Dr. Jörn Behrens"));
  xmlNewChild (resparty, NULL, BAD_CAST "organisationName", BAD_CAST "AWI Bremerhaven");
  xmlNewChild (resparty, NULL, BAD_CAST "role", BAD_CAST "principalInvestigator");
  xmlNewChild (citation, NULL, BAD_CAST "presentationForm", BAD_CAST "modelDigital");
  xmlNewChild (citation, NULL, BAD_CAST "otherCitationDetails",
    BAD_CAST "Projekt: German-Indonesian Tsunami Early Warning System");
  xmlNewChild (ident, NULL, BAD_CAST "abstract",
    xmlEncodeEntitiesReentrant (NULL, BAD_CAST "völlig freier Text\n> 1 Zeile"));
  xmlNewChild (ident, NULL, BAD_CAST "language", BAD_CAST "eng");
  xmlNewChild (ident, NULL, BAD_CAST "topicCategory", BAD_CAST "oceans");
  xmlNewChild (ident, NULL, BAD_CAST "extent", BAD_CAST NULL);
  xmlNewChild (root_node, NULL, BAD_CAST "characterSet", BAD_CAST "usAscii");
  xmlNewChild (root_node, NULL, BAD_CAST "dateStamp", BAD_CAST "2006-10-01");
  xmlNewChild (root_node, NULL, BAD_CAST "metadataStandardName",
    BAD_CAST "ISO 19115 Geographic information - Metadata");
  xmlNewChild (root_node, NULL, BAD_CAST "metadataStandardVersion", BAD_CAST "ISO 19115:2003");
  xmlSaveFormatFileEnc ("-", doc, "ASCII", 1);
  xmlFreeDoc (doc);
  exit (0); }
```

```
<?xml version="1.0" encoding="ASCII"?>
<MD_Metadata>
  <identificationInfo>
    <MD_DataIdentification>
      <citation>
        <CI_Citation>
          <title>Tsunami Simulation</title>
          <CI_ResponsibleParty>
            <individualName>
              PD Dr. J&#xF6;rn Behrens
            </individualName>
            <organisationName>AWI Bremerhaven
            </organisationName>
            <role>principalInvestigator</role>
          </CI_ResponsibleParty>
          <presentationForm>modelDigital
          </presentationForm>
          <otherCitationDetails>
            Projekt: German-Indonesian Tsunami
            Early Warning System
          </otherCitationDetails>
        </CI_Citation>
      </citation>
      <abstract>v&#xF6;llig freier Text
      &gt; 1 Zeile</abstract>
      <language>eng</language>
      <topicCategory>oceans</topicCategory>
      <extent/>
    </MD_DataIdentification>
  </identificationInfo>
  <characterSet>usAscii</characterSet>
  <dateStamp>2006-10-01</dateStamp>
  <metadataStandardName>
    ISO 19115 Geographic information - Metadata
  </metadataStandardName>
  <metadataStandardVersion>ISO 19115:2003
  </metadataStandardVersion>
</MD_Metadata>
```

Programm zur Erzeugung ISO 19115 - konformer Metadatenfiles



Dateistatus und Zeitstempel Abfragen (betriebssystemabhängig!)

Aktueller Zeitstempel:

```
#include <time.h>
```

```
time_t currenttime;  
char currentdatestring[11];
```

```
currenttime = time (NULL);  
strftime (currentdatestring,  
    (size_t)11, "%Y-%m-%d", localtime  
    (&currenttime));
```

Dateigröße und -zeitstempel:

```
#include <sys/types.h>
```

```
#include <sys/stat.h>
```

```
#include <time.h>
```

```
#include <stdio.h>
```

```
struct stat ls;  
char *inputfilename, mdatestring[11], size;
```

```
stat (inputfilename, &ls)  
strftime (mdatestring,  
    (size_t)11, "%Y-%m-%d", localtime (&ls.st_mtime));  
sprintf (size, "%i", (int)ls.st_size);
```


Erläuterung: Einfügen des aktuellen Zeitstempels

```
#include <libxml/xmlwriter.h>
#include <sys/types.h>
#include <time.h>
#include <unistd.h>

int main ()
{ xmlDocPtr doc;
  xmlNodePtr root_node, ident, resparty, dataidinfo, citation, idcitation;
  time_t currenttime; char currentdatestring[11];

  currenttime = time (NULL);
  strftime (currentdatestring, (size_t)11, "%Y-%m-%d", localtime (&currenttime));
  doc = xmlNewDoc (BAD_CAST "1.0");
  root_node = xmlNewNode (NULL, BAD_CAST "MD_Metadata");
  xmlDocSetRootElement (doc, root_node);
  dataidinfo = xmlNewChild (root_node, NULL, BAD_CAST "identificationInfo", BAD_CAST NULL);
  ident = xmlNewChild (dataidinfo, NULL, BAD_CAST "MD_DataIdentification", BAD_CAST NULL);
  idcitation = xmlNewChild (ident, NULL, BAD_CAST "citation", BAD_CAST NULL);
  citation = xmlNewChild (idcitation, NULL, BAD_CAST "CI_Citation", BAD_CAST NULL);
  xmlNewChild (citation, NULL, BAD_CAST "title", BAD_CAST "Tsunami Simulation");
  resparty = xmlNewChild (citation, NULL, BAD_CAST "CI_ResponsibleParty", BAD_CAST NULL);
  xmlNewChild (resparty, NULL, BAD_CAST "individualName",
    xmlEncodeEntitiesReentrant (NULL, BAD_CAST "PD. Dr. Jörn Behrens"));
  xmlNewChild (resparty, NULL, BAD_CAST "organisationName", BAD_CAST "AWI Bremerhaven");
  xmlNewChild (resparty, NULL, BAD_CAST "role", BAD_CAST "principalInvestigator");
  xmlNewChild (citation, NULL, BAD_CAST "presentationForm", BAD_CAST "modelDigital");
  xmlNewChild (citation, NULL, BAD_CAST "otherCitationDetails",
    BAD_CAST "Projekt: German-Indonesian Tsunami Early Warning System");
  xmlNewChild (ident, NULL, BAD_CAST "abstract",
    xmlEncodeEntitiesReentrant (NULL, BAD_CAST "völlig freier Text\n> 1 Zeile"));
  xmlNewChild (ident, NULL, BAD_CAST "language", BAD_CAST "eng");
  xmlNewChild (ident, NULL, BAD_CAST "topicCategory", BAD_CAST "oceans");
  xmlNewChild (ident, NULL, BAD_CAST "extent", BAD_CAST NULL);
  xmlNewChild (root_node, NULL, BAD_CAST "characterSet", BAD_CAST "usAscii");
  xmlNewChild (root_node, NULL, BAD_CAST "dateStamp", BAD_CAST currentdatestring);
  xmlNewChild (root_node, NULL, BAD_CAST "metadataStandardName",
    BAD_CAST "ISO 19115 Geographic information - Metadata");
  xmlNewChild (root_node, NULL, BAD_CAST "metadataStandardVersion", BAD_CAST "ISO 19115:2003");
  xmlSaveFormatFileEnc ("-", doc, "ASCII", 1);
  xmlFreeDoc (doc);
  exit (0); }
```

```
<?xml version="1.0" encoding="ASCII"?>
<MD_Metadata>
  <identificationInfo>
    <MD_DataIdentification>
      <citation>
        <CI_Citation>
          <title>Tsunami Simulation</title>
          <CI_ResponsibleParty>
            <individualName>
              PD Dr. J&#xF6;rn Behrens
            </individualName>
            <organisationName>AWI Bremerhaven
            </organisationName>
            <role>principalInvestigator</role>
          </CI_ResponsibleParty>
          <presentationForm>modelDigital
          </presentationForm>
          <otherCitationDetails>
            Projekt: German-Indonesian Tsunami
            Early Warning System
          </otherCitationDetails>
        </CI_Citation>
      </citation>
      <abstract>v&#xF6;llig freier Text
      &gt; 1 Zeile</abstract>
      <language>eng</language>
      <topicCategory>oceans</topicCategory>
      <extent/>
    </MD_DataIdentification>
  </identificationInfo>
  <characterSet>usAscii</characterSet>
  <dateStamp>2006-10-01</dateStamp>
  <metadataStandardName>
    ISO 19115 Geographic information - Metadata
  </metadataStandardName>
  <metadataStandardVersion>ISO 19115:2003
  </metadataStandardVersion>
</MD_Metadata>
```

URIs (Uniform Resource Identifier)

Allgemeine Form einer URI:

<Schema>://[<Benutzer>[:<Passwort>]@]<Server>[:<Port>]/[<Pfad>][?<Anfrage>][#<Fragment>]

Struktur einer URI in libxml2:

```
typedef struct _xmlURI xmlURI;  
typedef xmlURI *xmlURIPtr;  
struct _xmlURI {  
    char *scheme; /* the URI scheme */  
    char *opaque; /* opaque part */  
    char *authority; /* the authority part */  
    char *server; /* the server part */  
    char *user; /* the user part */  
    int port; /* the port number */  
    char *path; /* the path string */  
    char *query; /* the query string */  
    char *fragment; /* the fragment identifier */  
    int cleanup; /* parsing potentially unclean URI */  
};
```

Programmbeispiel:

```
xmlURIPtr uri;
```

```
uri = xmlCreateURI();  
uri->scheme = "http";  
uri->server = "www.unigis.ac.at";  
uri->path = "/club/u2/2006/Sonntag01.10.2006.html";  
xmlSaveUri (uri);
```

Ergebnis:

```
http://www.unigis.ac.at/club/u2/2006/Sonntag01.10.2006.html
```

Beispielprogramm URIs (Struktur, Vereinfachen, Encoding)

```
#include <libxml/xmlwriter.h>
#include <libxml/uri.h>
#include <unistd.h>

int main (int argc, char **argv)
{ xmlDocPtr doc;
  xmlNodePtr root_node, ident, resparty, dataidinfo, citation, idcitation,
  distributioninfo, distribution, transferoptions, digitaltransferoptions, online, onlinesrc;
  xmlURIPtr uri;

  uri = xmlCreateURI ();
  uri->scheme= "file";
  uri->path = argv[1];
  xmlNormalizeURIPath (uri->path);
  doc = xmlNewDoc (BAD_CAST "1.0");
  root_node = xmlNewNode (NULL, BAD_CAST "MD_Metadata");
  xmlDocSetRootElement (doc, root_node);
  dataidinfo = xmlNewChild (root_node, NULL, BAD_CAST "identificationInfo", BAD_CAST NULL);
  ident = xmlNewChild (dataidinfo, NULL, BAD_CAST "MD_DataIdentification", BAD_CAST NULL);
  idcitation = xmlNewChild (ident, NULL, BAD_CAST "citation", BAD_CAST NULL);
  citation = xmlNewChild (idcitation, NULL, BAD_CAST "CI_Citation", BAD_CAST NULL);
  xmlNewChild (citation, NULL, BAD_CAST "title", BAD_CAST "Tsunami Simulation");
  resparty = xmlNewChild (citation, NULL, BAD_CAST "CI_ResponsibleParty", BAD_CAST NULL);
  xmlNewChild (resparty, NULL, BAD_CAST "individualName",
  xmlEncodeEntitiesReentrant (NULL, BAD_CAST "PD. Dr. Jörn Behrens"));
  xmlNewChild (resparty, NULL, BAD_CAST "organisationName", BAD_CAST "AWI Bremerhav");
  xmlNewChild (resparty, NULL, BAD_CAST "role", BAD_CAST "principalInvestigator");
  xmlNewChild (citation, NULL, BAD_CAST "presentationForm", BAD_CAST "modelDigital");
  xmlNewChild (citation, NULL, BAD_CAST "otherCitationDetails",
  BAD_CAST "Projekt: German-Indonesian Tsunami Early Warning System");
  distributioninfo = xmlNewChild (root_node, NULL, BAD_CAST "distributionInfo", BAD_CAST NULL);
  distribution = xmlNewChild (distributioninfo, NULL, BAD_CAST "MD_Distribution", BAD_CAST NU);
  transferoptions = xmlNewChild (distribution, NULL, BAD_CAST "transferOptions", BAD_CAST NU);
  digitaltransferoptions = xmlNewChild (transferoptions, NULL, BAD_CAST "MD_DigitalTransferOpt
  NULL);
  online = xmlNewChild (digitaltransferoptions, NULL, BAD_CAST "onLine", BAD_CAST NULL);
  onlinesrc = xmlNewChild (online, NULL, BAD_CAST "CI_OnlineSrc", BAD_CAST NULL);
  xmlNewChild (digitaltransferoptions, NULL, BAD_CAST "CI_OnlineSrc", xmlSaveUri (uri));
  xmlNewChild (root_node, NULL, BAD_CAST "metadataStandardName",
  BAD_CAST "ISO 19115 Geographic information - Metadata");
  xmlSaveFormatFileEnc ("- ", doc, "ASCII", 1);
  xmlFreeDoc (doc);
  exit (0); }
```

```
$ createmeta "/home/braune/daten/./daten/netcdf test.nc"
<?xml version="1.0" encoding="ASCII"?>
<MD_Metadata>
  <identificationInfo>
    <MD_DataIdentification>
      <citation>
        <CI_Citation>
          <title>Tsunami Simulation</title>
          <CI_ResponsibleParty>
            <individualName>
              PD Dr. J&#xF6;r;n Behrens
            </individualName>
            <organisationName>AWI Bremerhaven
            </organisationName>
            <role>principalInvestigator</role>
          </CI_ResponsibleParty>
          <presentationForm>modelDigital</presentationForm>
          <otherCitationDetails>Projekt: German-Indonesian Tsunami Early
            Warning System</otherCitationDetails>
        </CI_Citation>
      </citation>
    </MD_DataIdentification>
  </identificationInfo>
  <distributionInfo>
    <MD_Distribution>
      <transferOptions>
        <MD_DigitalTransferOptions>
          <onLine>
            <CI_OnlineSrc>
              <linkage>file:///home/braune/daten/netcdf/%20test.nc
            </linkage>
            </CI_OnlineSrc>
          </onLine>
        </MD_DigitalTransferOptions>
      </transferOptions>
    </MD_Distribution>
  </distributionInfo>
  <metadataStandardName>
    ISO 19115 Geographic information - Metadata
  </metadataStandardName>
</MD_Metadata>
```


Erläuterung: Kopieren eines kompletten Teilbaums

Manchmal sinnvoll, z.B. bei Datenquellen und Prozessierungsschritten:

```
<?xml version="1.0" encoding="ASCII" ?>
<MD_Metadata>
  <dataQualityInfo>
    <DQ_DataQuality>
      <lineage>
        <LI_Lineage>
          <source><!-- GEBCO Ocean Bottom 1' Model --><source>
          <source><!-- SRTM Data (Land) 3" --><source>
          <source><!-- Ocean Bottom Data from FS Sonne 400m --><source>
          <source><!-- Initial Ocean Bottom vertical Displacement --><source>
          <processStep>
            <LI_ProcessStep>
              <description>Creation of Finite Elemente Lattice</description>
              <source><!-- GEBCO Ocean Bottom 1' Model --><source>
              <source><!-- SRTM Data (Land) 3" --><source>
              <source><!-- Ocean Bottom Data from FS Sonne 400m --><source>
            </LI_ProcessStep>
          </processStep>
          <processStep>
            <LI_ProcessStep>
              <description>Simulation on Finite Elemente Lattice with Initial Conditions</description>
              <source><!-- GEBCO Ocean Bottom 1' Model --><source>
              <source><!-- SRTM Data (Land) 3" --><source>
              <source><!-- Ocean Bottom Data from FS Sonne 400m --><source>
              <source><!-- Initial Ocean Bottom vertical Displacement --><source>
            </LI_ProcessStep>
          </processStep>
        </LI_Lineage>
      </lineage>
    </DQ_DataQuality>
  </MD_Metadata>
```

Kopieren eines kompletten Teilbaums (Beispiel: Bounding Box)

```
#include <libxml/xmlwriter.h>
#include <unistd.h>

int main ()
{ xmlDocPtr doc;
  xmlNodePtr root_node, dataidinfo, ident, idcitation, citation, resparty,
    extent, extent_, geographicElement, geobndbox, dqinfo, dataqual, scope, scope_;

doc = xmlNewDoc (BAD_CAST "1.0");
root_node = xmlNewNode (NULL, BAD_CAST "MD_Metadata");
xmlDocSetRootElement (doc, root_node);
dataidinfo = xmlNewChild (root_node, NULL, BAD_CAST "identificationInfo", BAD_CAST NULL);
ident = xmlNewChild (dataidinfo, NULL, BAD_CAST "MD_DataIdentification", BAD_CAST NULL);
idcitation = xmlNewChild (ident, NULL, BAD_CAST "citation", BAD_CAST NULL);
citation = xmlNewChild (idcitation, NULL, BAD_CAST "CI_Citation", BAD_CAST NULL);
xmlNewChild (citation, NULL, BAD_CAST "title", BAD_CAST "Tsunami Simulation");
resparty = xmlNewChild (citation, NULL, BAD_CAST "CI_ResponsibleParty", BAD_CAST NULL);
xmlNewChild (resparty, NULL, BAD_CAST "individualName",
  xmlEncodeEntitiesReentrant (NULL, BAD_CAST "PD. Dr. Jörn Behrens"));
extent = xmlNewChild (ident, NULL, BAD_CAST "extent", BAD_CAST NULL);
extent_ = xmlNewChild (extent, NULL, BAD_CAST "EX_Extent", BAD_CAST NULL);
geographicElement = xmlNewChild (extent_, NULL, BAD_CAST "geographicElement", BAD_CAST NULL);
geobndbox = xmlNewChild (geographicElement, NULL, BAD_CAST "EX_GeographicBoundingBox", BAD_CAST
  xmlNewChild (geobndbox, NULL, BAD_CAST "westBoundLongitude", BAD_CAST "77.38");
  xmlNewChild (geobndbox, NULL, BAD_CAST "eastBoundLongitude", BAD_CAST "128.09");
  xmlNewChild (geobndbox, NULL, BAD_CAST "southBoundLatitude", BAD_CAST "-30.00");
  xmlNewChild (geobndbox, NULL, BAD_CAST "northBoundLatitude", BAD_CAST "25.36");
dqinfo = xmlNewChild (root_node, NULL, BAD_CAST "dataQualityInfo", BAD_CAST NULL);
dataqual = xmlNewChild (dqinfo, NULL, BAD_CAST "DQ_DataQuality", BAD_CAST NULL);
scope = xmlNewChild (dataqual, NULL, BAD_CAST "scope", BAD_CAST NULL);
scope_ = xmlNewChild (scope, NULL, BAD_CAST "DQ_Scope", BAD_CAST NULL);
xmlNewChild (scope_, NULL, BAD_CAST "level", BAD_CAST "model");
xmlAddChild (scope_, extent);
xmlNewChild (root_node, NULL, BAD_CAST "metadataStandardName",
  BAD_CAST "ISO 19115 Geographic information - Metadata");
xmlSaveFormatFileEnc ("-", doc, "ASCII", 1);
xmlFreeDoc (doc); exit (0); }
```

```
<?xml version="1.0" encoding="ASCII"?>
<MD_Metadata>
  <identificationInfo>
    <MD_DataIdentification>
      <citation>
        <CI_Citation>
          <title>Tsunami Simulation</title>
          <CI_ResponsibleParty>
            <individualName>
              PD Dr. J&#xF6;r;n Behrens
            </individualName>
          </CI_Citation>
        </citation>
      </MD_DataIdentification>
    </identificationInfo>
    <dataQualityInfo>
      <DQ_DataQuality>
        <scope>
          <DQ_Scope>
            <level>model</level>
          </DQ_Scope>
        </scope>
      </DQ_DataQuality>
    </dataQualityInfo>
    <metadataStandardName>
      ISO 19115 Geographic information - Metadata
    </metadataStandardName>
  </MD_Metadata>
```

Encoding UTF-8, UTF-16, ISO-8859-1, ASCII, HTML

xmlSaveFormatFileEnc (const char *filename, xmlDocPtr cur, "UTF-8" , int format);	<characterSet> utf8 </characterSet>	PD Dr. <u>J</u> <u>M</u> - <u>C</u> <u>M</u> - <u>6</u> <u>r</u> n Behrens PD Dr. <u>J</u> <u>A</u> <u>r</u> n Behrens	file:///data/tsunami% <u>20</u> java% <u>20</u> triple0.nc
xmlSaveFormatFileEnc (const char *filename, xmlDocPtr cur, "UTF-16" , int format);	<characterSet> utf16 </characterSet>	PD Dr. <u>J</u> <u>M</u> - <u>v</u> <u>r</u> n Behrens PD Dr. <u>J</u> <u>o</u> <u>r</u> n Behrens	file:///data/tsunami% <u>20</u> java% <u>20</u> triple0.nc
xmlSaveFormatFileEnc (const char *filename, xmlDocPtr cur, "ISO-8859-1" , int format);	<characterSet> 8859part1 </characterSet>	PD Dr. <u>J</u> <u>o</u> <u>r</u> n Behrens	file:///data/tsunami% <u>20</u> java% <u>20</u> triple0.nc
xmlSaveFormatFileEnc (const char *filename, xmlDocPtr cur, "ASCII" , int format);	<characterSet> usAscii </characterSet>	PD Dr. <u>J</u> <u>&#246</u> ;r n Behrens	file:///data/tsunami% <u>20</u> java% <u>20</u> triple0.nc
xmlSaveFormatFileEnc (const char *filename, xmlDocPtr cur, "HTML" , int format);	<characterSet> usAscii </characterSet>	PD Dr. <u>J</u> <u>&ouml</u> ;r n Behrens	file:///data/tsunami% <u>20</u> java% <u>20</u> triple0.nc